KOLLMORGEN

www.DanaherMotion.com

Kommunikationsprofil CANopen für SERVOSTAR®



Technische Beschreibung, Inbetriebnahmeanleitung



Ausgabe 09.02
Datei srcan_d.xxx
EU Mat.Nr..: 90809

Bisher erschienene Ausgaben

Ausgabe	Bemerkung
01 / 99	Erstausgabe, gültig ab Software-Versionen 1.57
08 / 99	Layout, kleinere Korrekturen
03 / 01	Ergänzung des Objektverzeichnisses, gültig ab Firmwareversion 4.20
08 / 01	Detail-Korrekturen, Index erweitert, Objektverzeichnis erweitert, gültig ab Firmwareversion 4.80
09 / 02	neues Layout, TPDO 34 neu, diverse Korrekturen

SERVO**STAR** ist ein eingetragenes Warenzeichen der Kollmorgen Corporation

Technische Änderungen, die der Verbesserung der Geräte dienen, vorbehalten!

Gedruckt in der BRD

Seite

Seite

IV CANopen Antriebsprofil

Seite

Seite

VI Anhang

In diesem Handbuch verwendete Kürzel

In der Tabelle unten werden die in diesem Handbuch verwendeten Abkürzungen erklärt.

Kürzel	Bedeutung
BTB/RTO	Betriebsbereit
COB-ID	Communication Object Identifier
EEPROM	Elektrisch löschbarer Festspeicher
EMV	Elektromagnetische Verträglichkeit
ISO	International Standardization Organisation
LED	Leuchtdiode
MB	Megabyte
NSTOP	Endschaltereingang Drehrichtung links
PC	Personal Computer mit 80x86-Prozessor
PDO	Prozessdatenobjekt

Kürzel	Bedeutung
PSTOP	Endschaltereingang Drehrichtung rechts
RAM	flüchtiger Speicher
RES	Resolver
ROD	Inkrementelle Positionsausgabe
RPDO	Receive- (Empfangs-) PDO
SDO	Servicadatenobjekt
SPS	Speicherprogrammierbare Steuerung
SW/SETP.	Sollwert (setpoint)
TPDO	Transmit- (Sende-) PDO

In diesem Handbuch verwendete Symbole

Symbol	Bedeutung		Symbol	Bedeutung
⚡	personelle Gefährdung durch Elektrizität und ihre Wirkung		⚠	Allgemeine Warnung Allgemeine Hinweise maschinelle Gefährdung
⇨ S.	siehe Seite (Querverweis)		●	Hervorhebung

Diese Seite wurde bewusst leer gelassen.

I Allgemeines

I.1 Über dieses Handbuch

Dieses Handbuch beschreibt Inbetriebnahme, Funktionsumfang und Softwareprotokoll des Servoverstärkers SERVO**STAR**™ 400/600 mit dem Kommunikationsprofil CANopen.
Es ist Bestandteil der Gesamtdokumentation der digitalen Servoverstärker-Familie SERVOSTAR.

Installation und Inbetriebnahme der Servoverstärker, sowie alle Standardfunktionen werden in der zugehörigen Installationsanleitung beschrieben.

Sonstige Bestandteile der Gesamtdokumentation der digitalen Servoverstärker-Familien:

Titel	Herausgeber
Handbuch Inbetriebnahmesoftware DRIVE.EXE	Kollmorgen
Montage-/ Installations-/ Inbetriebnahmeanleitung SERVOSTAR	Kollmorgen

Weiterführende Dokumentation:

Titel	Herausgeber
CAN Application Layer (CAL) for Industrial Applications	CiA e.V.
Draft standards 301 (ab Version 4.0), 401	CiA e.V.
CAN Specification Version 2.0	CiA e.V.
ISO 11898 ...Controller area network (CAN) for high-speed communication	

Dieses Handbuch richtet sich mit folgenden Anforderungen an Fachpersonal:



Verdrahtung: *Fachleute mit elektrotechnischer Ausbildung*
Programmierung: *Softwareentwickler, CAN-BUS Projekteure*

Wir bieten auf Anfrage Schulungs- und Einarbeitungskurse an.

I.2 Verwendung dieses Handbuches

Konkrete Beispiele zu einzelnen Kapiteln sind im Applikationsteil (⇒ VI.1 und VI.2) dieses Handbuches zu finden.

I.3 Bestimmungsgemäße Verwendung des CANopen Interfaces

Beachten Sie das Kapitel "Bestimmungsgemäße Verwendung" im Inbetriebnahmehandbuch des Servoverstärkers.

Das Interface ist Bestandteil der digitalen Servoverstärker der Serie SERVOSTAR.
Das CANopen Interface dient allein dem Anschluss des Servoverstärkers an einen Master mit CAN BUS Anbindung.
Die Servoverstärker werden als Komponenten in elektrischen Anlagen oder Maschinen eingebaut und dürfen nur als integrierte Komponenten der Anlage in Betrieb genommen werden.



Wir garantieren nur bei Verwendung der von uns genannten Komponenten und Einhaltung der Installationsvorschriften die Konformität der Servoverstärker zu folgenden Normen im Industriebereich:

EG-EMV-Richtlinie	***89/336/EWG***
EG-Niederspannungs-Richtlinie	***73/23/EWG***

I.4 Realisierte Grundfunktionen über CANopen

In Zusammenhang mit dem im digitalen Servoverstärker SERVOSTAR integrierten Lageregler werden folgende Funktionen bereitgestellt:

Einricht- und Allgemeine Funktionen:

— Referenzfahren, Referenzpunkt setzen
— Tippen mit variabler Geschwindigkeit
— Vorgabe digitaler Sollwerte für die Drehzahl- und Momentenregelung

Positionierfunktionen:

— Ausführen eines Fahrauftrages aus dem Fahrsatzspeicher des Servoverstärkers
— Ausführen eines Direktfahrauftrages
— Trajektorie absolut

Datentransferfunktionen:

— Übertragen eines Fahrauftrages in den Fahrsatzspeicher des Servoverstärkers
Ein Fahrauftrag besteht aus folgenden Elementen:
» Positionssollwert (Absolutauftrag) oder Wegsollwert (Relativauftrag)
» Geschwindigkeitssollwert
» Beschleunigungszeit, Bremszeit, bei Bedarf Ruckbegrenzungszeiten (i.V.)
» Fahrauftragsart (absolut/relativ)
» Nummer eines Folgefahrauftrags (mit oder ohne Zwischenstopp)
— Lesen eines Fahrauftrages aus dem Fahrsatzspeicher des Servoverstärkers
— Lesen von Istwerten
— Lesen der Fehlerregister
— Lesen der Statusregister
— Lesen / Schreiben von Regelparametern

I.5 Systemvoraussetzungen

— Servoverstärker SERVOSTAR
— Masterstation mit CAN-Bus Anbindung (z.B. PC mit CAN-Interface)

I.6 Übertragungsgeschwindigkeit und -verfahren

— Busankopplung und Busmedium: CAN-Standard ISO 11898 (CAN-Highspeed)
— Übertragungsgeschwindigkeit: max. 1MBit/s
Einstellmöglichkeiten des Servoverstärkers:
10, 20, 50, 100, 125, 250, 333, 500(default), 666, 800, 1000kBaud

I.7 Busleitung

Nach ISO 11898 sollten Sie eine Busleitung mit einem Wellenwiderstand von 120 Ω verwenden. Die verwendbare Leitungslänge für eine sichere Kommunikation nimmt mit zunehmender Übertragungsrate ab. Als Anhaltspunkte können folgende bei uns gemessenen Werte dienen, die allerdings nicht als Grenzwerte zu verstehen sind:

Leitungsdaten: Wellenwiderstand 100..120 Ω
Leitungskapazität max. 60 nF/km
Leiterwiderstand (Schleife) 159,8 Ω/km

Leitungslängen in Abhängigkeit von der Übertragungsrate

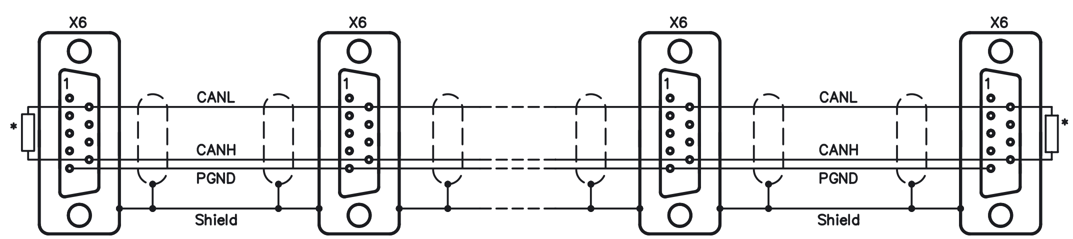
Übertragungsrate / kBaud	max. Leitungslänge / m
1000	20
500	70
250	115

Mit geringerer Leitungskapazität (max. 30 nF/km) und geringerem Leiterwiderstand (Schleife, 115 Ω/km) können größere Übertragungsweiten erreicht werden.

(Wellenwiderstand 150 ± 5Ω ⇒ Abschlusswiderstand 150 ± 5Ω).

An das SubD-Steckergehäuse sind aus EMV-Gründen folgende Anforderung gestellt:
— metallisches oder metallisch beschichtetes Gehäuse
— Anschlussmöglichkeit für den Leitungsschirm im Gehäuse, großflächige Verbindung

Von der Firma Danaher Motion können spezielle Klemmbuchsenstecker (Best.Nr. 90650) bezogen werden, die für den Busbetrieb leicht konfektioniert werden können. Darüber hinaus kann auch ein CAN – Bus – Adapter für den Optionsschacht verwendet werden (Option -2CAN-, Best.Nr. 101174), der die Möglichkeit der Durchverdrahtung und das optionale Zuschalten des 120Ω Terminierungswiderstandes bietet. Die Pinbelegung entspricht dem Standard CANopen DS301.



I.8 Verhalten bei Kommunikationsstörungen BUSSOFF

Die Kommunikationsstörung BUSOFF wird direkt von der Schicht 2 (CAN controller) überwacht und gemeldet. Diese Meldung kann unterschiedliche Ursachen haben.

Hier einige Beispiele:
- Telegramm wird gesendet, obwohl kein weiterer CAN – Knoten angeschlossen ist
- CAN Knoten weisen unterschiedliche Baudraten auf
- Busleitung defekt
- Reflexionen auf den Leitungen aufgrund fehlerhafter Leitungsabschlüsse.

Ein BUSOFF wird vom SERVOSTAR nur gemeldet, wenn ein weiterer CAN Knoten angeschlossen ist und mindesten ein Objekt zu Beginn erfolgreich abgesetzt werden konnte. Der Zustand BUSOFF wird mit der Fehlermeldung F23 signalisiert. Sollte beim Auftreten dieses Fehlers die Endstufe freigegeben (enable) sein und eine Fahrfunktion ausgeführt werden, wird der Antrieb mit der Notbremsrampe angehalten und die Endstufe gesperrt (disable).

Diese Seite wurde bewusst leer gelassen

II Installation / Inbetriebnahme

II.1 Montage, Installation



Installieren und verdrahten Sie die Geräte immer in spannungsfreiem Zustand. Weder die Leistungsversorgung, noch die 24V-Hilfsspannung, noch die Betriebsspannung eines anderen anzuschliessenden Gerätes darf eingeschaltet sein.
Sorgen Sie für eine sichere Freischaltung des Schaltschrankes (Sperre, Warnschilder etc.). Erst bei der Inbetriebnahme werden die einzelnen Spannungen eingeschaltet.
Lösen Sie die elektrischen Anschlüsse der Servoverstärker nie unter Spannung. Es könnte zu Zerstörungen der Elektronik kommen.
Restladungen in den Kondensatoren können auch mehrere Minuten nach Abschalten der Netzspannung gefährliche Werte aufweisen. Messen Sie die Spannung im Zwischenkreis und warten Sie, bis die Spannung unter 40V abgesunken ist.
Steuer- und Leistungsanschlüsse können Spannung führen, auch wenn sich der Motor nicht dreht.

- Montieren Sie den Servoverstärker wie in der Installationsanleitung SERVOSTAR beschrieben. Beachten Sie alle Sicherheitshinweise in der zum Servoverstärker gehörenden Installationsanleitung. Beachten Sie alle Hinweise zu Einbaulage, Umgebungsbedingungen und Verdrahtung, sowie Absicherung.
- Den Motor-, Steuerungs- und Leistungsanschluss, sowie Hinweise zu EMV-gerechtem Systemaufbau finden Sie in der Installationsanleitung des Servoverstärkers.

II.1.1 Anschlusstechnik

Netzanschluss, Motoranschluss, Analoge Sollwerte, Digitale Steuersignale, CAN-Anschluss siehe Installationsanleitung SERVOSTAR.

II.1.2 Einstellen der Stationsadresse

Die Stationsadresse (Geräteadresse am CAN-Bus) des Servoverstärkers können Sie auf drei Arten eingestellen:

- Mit der Tastatur in der Frontplatte (siehe Installationsanleitung SERVOSTAR)
- In der Inbetriebnahme-Software DRIVE.EXE auf der Bildschirmseite “Basiseinstellungen”
- Über die serielle Schnittstelle mit der Abfolge der ASCII-Kommandos:
  ADDR nn ⇒ SAVE ⇒ COLDSTART (mit nn = Adresse)

Der Adressbereich kann mit Hilfe des ASCII – Objektes MDRV von 1..63 auf 1..127 expandiert werden.

II.1.3 Einstellen der Baudrate

Die CAN - Übertragungsgeschwindigkeit (Baudrate) können Sie auf drei Arten eingestellen:

- Mit der Tastatur in der Frontplatte (siehe Installationsanleitung SERVOSTAR)
- In der Inbetriebnahme-Software DRIVE.EXE auf der Bildschirmseite “Basiseinstellungen”
- Über die serielle Schnittstelle mit der Abfolge der ASCII - Kommandos:
  CBAUD bb ⇒ SAVE ⇒ COLDSTART (mit bb = Baudrate in kB)

Mögliche Baudraten sind 10, 20, 50, 100, 125, 250, 333, 500(default), 666, 800, 1000kBaud.

II.2 Inbetriebnahme



Nur Fachpersonal mit fundierten Kenntnissen in Regelungstechnik und Antriebstechnik darf den Servoverstärker in Betrieb nehmen.

Schritt	Beschreibung
Montage / Installation prüfen	Prüfen Sie, ob alle Sicherheitshinweise in der Installationsanleitung des Servoverstärkers und im vorliegenden Handbuch beachtet und umgesetzt wurden. Kontrollieren Sie die eingestellte Stationsadresse.
PC anschließen, Inbetriebnahmesoftware starten	Zum Parametrieren des Servoverstärkers verwenden Sie die Inbetriebnahmesoftware DRIVE.EXE.
Grundfunktionen in Betrieb nehmen	Nehmen Sie nun die Grundfunktionen des Servoverstärkers in Betrieb und optimieren Sie Strom- und Drehzahlregler. Dieser Teil der Inbetriebnahme ist im Handbuch der Inbetriebnahmesoftware genauer beschrieben.
Parameter speichern	Speichern Sie die Parameter nach erfolgter Optimierung im Servoverstärker.
Buskommunikation in Betrieb nehmen	Die geänderten Parameter werden erst nach einem Software-Reset (Warmboot) wirksam. Wechseln Sie hierzu auf die Bildschirmseite “Status” und betätigen Sie den Reset-Button. Voraussetzung: das in Kapitel IV beschriebene Software-Protokoll ist auf dem Master realisiert. Passen Sie die Baudrate des SERVOSTAR an die des Masters an.
Test der Kommunikation	Vorschlag: Fordern Sie das Emergency-Objekt an.



Vorsicht!
Stellen Sie sicher, dass auch bei ungewollter Bewegung des Antriebs keine maschinelle oder personelle Gefährdung eintreten kann.

Schritt	Beschreibung
Lageregler in Betrieb nehmen	Nehmen Sie den Lageregler in Betrieb, wie im Handbuch der Inbetriebnahmesoftware beschrieben.

III CANopen Kommunikationsprofil

Dieses Kapitel beschreibt die grundlegenden Dienste und Kommunikationsobjekte des CANopen Kommunikationsprofils DS 301, die vom SERVOSTAR verwendet werden.
Es wird vorausgesetzt, dass das Kommunikationsprofil in seiner grundlegenden Funktionsweise bekannt ist und als Referenzdokumentation zur Verfügung steht.
Da die Geschäftssprache der CiA englisch ist, werden feststehende Begriffe im Original verwendet (z.B. Objekt 1000_h: Device Type).

III.1 Allgemeine Erläuterungen zu CAN

Das hier verwendete Übertragungsverfahren ist in der ISO 11898 (Controller Area Network [CAN] for high-speed communication) definiert.
Das in allen CAN-Bausteinen implementierte Schicht 1/2-Protokoll (Physical Layer/Data Link Layer) stellt u. a. die Anforderung von Daten zur Verfügung.

Datentransport bzw. Datenanforderung erfolgen über ein Datentelegramm (Data Frame) mit bis zu 8 Byte Nutzdaten bzw. über ein Datenanforderungstelegramm (Remote Frame).

Kommunikationsobjekte werden durch einen 11 Bit Identifier (ID) gekennzeichnet, der auch die Priorität von Objekten bestimmt.

Um die Applikation von der Kommunikation zu entkoppeln, wurde ein Schicht-7-Protokoll (Anwendungsschicht) entwickelt. Die von der Anwendungsschicht bereitgestellten Dienstelemente ermöglichen die Realisierung einer über das Netzwerk verteilten Applikation. Diese Dienstelemente sind im "CAN Applikation Layer (CAL) for Industrial Applications" beschrieben.
Auf die CAL ist das Kommunikationsprofil CANopen und das Antriebsprofil aufgesetzt.

Das folgende Diagramm zeigt die grundlegende Struktur eines Kommunikationsobjekts:

S O M	COB-ID	R T R	CTRL	Data Segment	CRC	A C K	EOM

SOM	Start of message
COB-ID	COB-Identifier (11 Bit)
RTR	Remote Transmission Request
CTRL	Control Field (u.a. Data Length Code)
Data Segment	0..8 Byte (Data-COB)
	0 Byte (Remote-COB)
CRC	Prüfsequenz
ACK	Acknowledge Slot
EOM	End of message

III.2 Aufbau des COB-Identifiers

Die folgende Grafik zeigt den Aufbau des COB – Identifiers (COB - ID). Der “Function Code” legt die Bedeutung und die Priorität des jeweiligen Objekts fest.

10	9	8	7	6	5	4	3	2	1	0
Code				Module-ID						

Bit 0..6 Modul ID (Stationsnummer, Bereich 1..63; wird in der Inbetriebnahmesoftware oder am Servoverstärker eingestellt, ⇒ II.1.2)

Bit 7..10 Function Code (Nummer des im Server definierten Kommunikationsobjektes)



Achtung: Wird eine ungültige Stationsnummer eingestellt (=0 oder >63), so wird die Modul-ID intern auf 1 gesetzt. Mit dem ASCII Objekt MDRV ist es möglich, den Adressbereich von 63 auf 127 zu expandieren.

Die folgenden Tabellen zeigen die Default - Belegung der COB - Identifier nach dem Einschalten des Servoverstärkers. Die Objekte, die mit einem Index (Communication Parameters at Index) versehen sind, können nach der Initialisierungsphase mit einem neuen Identifier versehen werden. Die Indizes in Klammern sind optional.

Vordefinierte “Broadcast” Objekte (Senden an alle)

Objekt	Function code (binary)	Resulting COB-IDs		Communication parameters at index
		Dez.	Hex.	
NMT	0000	0	0_h	—
SYNC	0001	128	80_h	(1005_h)
TIME	0010	256	100_h	—

Vordefinierte “Peer to Peer” Objekte (Senden Station zu Station)

Objekt	Function code (binary)	Resulting COB-IDs		Communication parameters at index	Piorität
		Dez.	Hex.		
EMERGENCY	0001	129..255	81_h..FF_h	—	hoch
TPDO 1	0011	385..511	181_h..$1FF_h$	1800_h	↓
RPDO 1	0100	513..639	201_h..$27F_h$	1400_h	
TPDO 2	0101	641..767	281_h..$2FF_h$	1801_h	
RPDO 2	0110	769..895	301_h..$37F_h$	1401_h	
TPDO 3	0110	897..1023	381_h..$3FF_h$	1802_h	
RPDO 3	1000	1025..1151	401_h..$47F_h$	1402_h	
TPDO 4	1001	1153..1279	481_h..$4FF_h$	1803_h	
RPDO 4	1010	1281..1407	501_h..$57F_h$	1403_h	
SDO (tx*)	1011	1409..1535	581_h..$5FF_h$		
SDO (rx*)	1100	1537..1663	601_h..$67F_h$		
Nodeguard	1110	1793..1919	701_h..$77F_h$	($100E_h$)	niedrig

* tx = Senderichtung SERVOSTAR ⇒ Master
rx = Senderichtung Master ⇒ SERVOSTAR

III.3 Defination der verwendeten Datentypen

Dieses Kapitel definiert die verwendeten Datentypen. Jeder Datentyp kann mit Hilfe von Bit -Sequenzen beschrieben werden. Diese Bit - Sequenzen werden in “Octets” (Bytes) zusammengefasst. Für numerische Datentypen wird das **“Little – Endian” – Format** (Intel) verwendet (s. auch DS301 Application Layer “General Description of Data Types and Encoding Rules”).

III.3.1 Basisdatentypen

III.3.1.1 Unsigned Integer

Daten vom Basis Datentyp UNSIGNEDn definiert ausschließlich positive Integer.
Der Wertebereich ist $0..2^n-1$. Die Bitsequenz $b = b_0..b_{n-1}$ definiert den Wert

$$\text{UNSIGNEDn}(b) = b_{n-1}\ 2^{n-1} +..+ b_1\ 2^1 + b_0\ 2^0$$

Beispiel: Der Wert 266 = $10A_h$ wird mit dem Datentyp UNSIGNED16 mit Hilfe von zwei “octets” übertragen (1. octet = $0A_h$, 2. octet = 01_h).

Übertragungssyntax für den Datentyp UNSIGNEDn:

octet Nummer	1.	2.	3.	4.	5.	6.	7.	8.
UNSIGNED8	$b_7..b_0$							
UNSIGNED16	$b_7..b_0$	$b_{15}..b_8$						
UNSIGNED24	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$					
UNSIGNED32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$				
UNSIGNED40	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$			
UNSIGNED48	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$		
UNSIGNED56	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	
UNSIGNED64	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	$b_{63}..b_{56}$

III.3.1.2 Signed Integer

Daten vom Basis Datentyp INTEGERn definiert positive und negative Integer.
Der Wertebereich ist $-2^{n-1}-1..2^{n-1}-1$. Die Bitsequenz $b = b_0..b_{n-1}$ definiert den Wert

$$\text{INTEGERn}(b) = b_{n-2}\ 2^{n-2} +..+ b_1\ 2^1 + b_0\ 2^0 \text{ mit } b_{n-1} = 0$$

Negative Zahlen werden im zweier Komplement dargestellt - somit gilt:

$$\text{INTEGERn}(b) = - \text{INTEGERn}(b) - 1 \text{ mit } b_{n-1} = 1$$

Beispiel: Der Wert -266 = $FEF6_h$ wird mit dem Datentyp INTEGER16 mit Hilfe von zwei “octets” übertragen (1. octet = $F6_h$, 2. octet = FE_h).

Übertragungssyntax für den Datentyp INTEGERn:

octet Nummer	1.	2.	3.	4.	5.	6.	7.	8.
INTEGER8	$b_7..b_0$							
INTEGER16	$b_7..b_0$	$b_{15}..b_8$						
INTEGER24	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$					
INTEGER32	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$				
INTEGER40	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$			
INTEGER48	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$		
INTEGER56	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	
INTEGER64	$b_7..b_0$	$b_{15}..b_8$	$b_{23}..b_{16}$	$b_{31}..b_{24}$	$b_{39}..b_{32}$	$b_{47}..b_{40}$	$b_{55}..b_{48}$	$b_{63}..b_{56}$

III.3.2 Zusammengesetzte Datentypen

Zusammengesetzte Datentypen nutzen Basisdatentypen (INTEGERn, UNSIGNEDn, REAL). Es werden zwei Datentypen unterschieden:

- STRUCT
  Dieser Datentyp setzt sich aus unterschiedlichen Basisdatentypen zusammen.
- ARRAY
  Dieser Datentyp setzt sich aus gleichen Basisdatentypen zusammen.

III.3.3 Erweiterte Datentypen

Erweiterte Datentypen werden aus Basisdatentypen und zusammengesetzten Datentypen abgeleitet. Im Folgenden werden die unterstützten Datentypen definiert.

III.3.3.1 Octet String

Der Datentyp “OCTET_STRING” definiert sich aus dem Datentyp ARRAY. “Length” ist die Länge des “octet string”.

ARRAY[length] OF UNSIGNED8	OCTET_STRINGlength

III.3.3.2 Visible String

Der Datentyp “VISIBLE_” kann sich aus dem Datentyp UNSIGNED8 oder aus dem Datentyp ARRAY definieren. Die zulässigen Werte sind 00_h und der Bereich von 20_h bis $7E_h$. Die Daten werden als 7 Bit ASCII Code interpretiert (ISO 646-1973(E)). “Length” ist die Länge des “visible string”.

UNSIGNED8	VISIBLE_CHAR
ARRAY[length] OF VISIBLE_CHAR	VISIBLE_STRINGlength

III.4 Kommunikationsobjekte

Die Kommunikationsobjekte werden mit Hilfe von Dienstelementen und Protokollen beschrieben. Es können im wesentlichen zwei Dienstelemente unterschieden werden:

- Unbestätigte Dienste
- Bestätigte Dienste

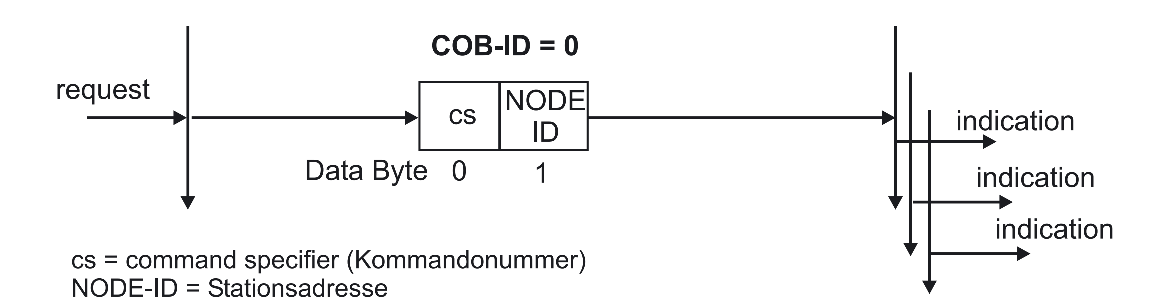
Sämtliche Dienste setzen voraus, das “Data Link” und “Physical Layer” fehlerfrei arbeiten.

Der SERVOSTAR unterstützt die Kommunikationsobjekte, die in den folgenden Kapiteln detailliert beschrieben werden:

- Network Management Objects (NMT)
- Synchronisation Object (SYNC)
- Time Stamp Object (TIME)
- Emergency Object (EMCY)
- Process Data Object (PDO)
- Service Data Object (SDO)
- Nodeguard

III.4.1 Network Management Objects (NMT)

Das NMT - Telegramm sieht wie folgt aus:



Der Antrieb unterstützt folgende Funktionen des Netzwerkmanagements:

cs = 129, reset node: führt zu einem Kaltstart des Antriebs, damit werden sämtliche im RAM abgelegte Parameter gelöscht und wieder die im EEPROM gespeicherten Werte bzw. Defaultwerte eingestellt.

cs = 1, start remote node: startet den CAN-Knoten, d.h. gibt die PDOs des Antriebs zur Bedienung frei. Ab diesem Zeitpunkt werden Sende-PDOs ereignisgesteuert gesendet, bzw. kann der zyklische Prozessdatenbetrieb starten.

cs = 2, Stop remote node: stoppt den CAN-Knoten, d.h. der Antrieb reagiert nicht mehr auf empfangene PDOs und sendet keine mehr.

III.4.2 Synchronisation Object (SYNC)

Das SYNC ist ein periodisches “Broadcast” – Objekt und gibt den Basis - Bustakt vor. Um einen zeitlich äquidistanten Abstand zu ermöglichen, besitzt das SYNC eine hohe Priorität. Die Anwendung dieses Protokolls wird im Applikationsteil dieses Handbuchs veranschaulicht.

III.4.3 Time Stamp Object (TIME)

Dieses Kommunikationsobjekt wird vom SERVOSTAR nicht unterstützt.

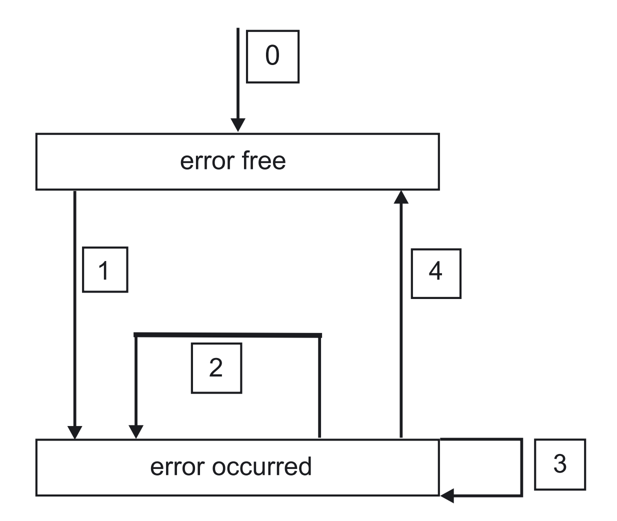
III.4.4 Emergency Object (EMCY)

Das EMCY wird ereignisgetriggert aufgrund einer internen Fehlersituation generiert. Für jeden Fehler wird dieses Objekt erneut übertragen. Da die Fehlercodes geräteabhängig sind, werden sie im Kapitel "CANopen Antriebsprofil" (⇒ Kap.IV) beschrieben.

III.4.4.1 Verwendung des Emergency Object

Das Verhalten im Fehlerfall ist von der Fehlerart abhängig und somit unterschiedlich. Aus diesem Grund wird das Verhalten mit Hilfe eine Fehlerzustandsmaschine beschrieben. Es werden die Fehlerzustände “error free” und “error occurred” unterschieden. Es sind die folgenden Übergänge definiert:

0. Nach der Initialisierung wird der Zustand “error free” eingenommen, falls kein Fehler erkannt wurde. In diesem Zustand wird keine Fehlermeldung generiert.
1. Der SERVOSTAR erkennt einen internen Fehler und zeigt diesen in den ersten drei Bytes des “emergency” Telegramms an (“error code” Byte 0,1 und “error register”Byte 2).
   Da der SERVOSTAR mehrere Fehlerarten unterscheidet, wird Byte 3 aus dem herstellerspezifischen Fehlerfeld dazu genutzt, um die Fehlerkategorie anzuzeigen.
2. Einer, aber nicht alle Fehler sind zurückgesetzt worden. Das EMCY Telegramm beinhaltet den “Error Code” 0000_h, das “Error Register” zeigt die restlichen noch anstehenden Fehler an. Der herstellerspezifische Bereich ist zu Null gesetzt.
3. Ein neuer Fehler ist aufgetreten. Der SERVOSTAR verbleibt im Zustand “Error State” und überträgt ein EMCY Objekt mit dem entsprechenden “Error Code”. Der neue “Error Code” wird in den Bytes 0 und 1 eingetragen.
4. Alle Fehler sind zurückgesetzt worden. Das EMCY Telegramm beinhaltet den “Error Code” 0000_h, das “Error Register” zeigt keine Fehler an.
   Der herstellerspezifische Bereich ist zu Null gesetzt.



III.4.4.2 Aufteilung des Emergency Object

Das Emergency - Objekt setzt sich aus 8 Byte zusammen und ist folgendermaßen aufgeteilt:

Byte	0	1	2	3	4	5	6	7
Inhalt	Emergency error code (⇒ IV.1)		Error register (Objekt 1001_h)	Kategorie	Reserviert			

Wurde ein Emergency Objekt generiert, wird im Anschluss daran der Fehlerzustand der Zustandsmaschine gemeldet (error free / error occured), indem ein zweites Emergency - Objekt generiert wird. Hier sind nur die ersten 4 Bytes relevant (Emergency ErrorCode , Error register, Kategorie). Byte 0/1 enthält den “Error Reset Code“ (0000_h) und Byte 2 zeigt an, dass eventuell noch ein Fehler ansteht. Ist das Error Register 00_h, ist der Fehlerzustand “error free“.
Byte 3 enthält die Kategorie. Die Bedeutung der Fehlernummern (Error Code) und der Fehlerkategorie werden im Kapitel “Emergency Message” (⇒ Kap. IV.1) beschrieben. Das Fehlerregister wird über das Objekt 1001_h “Error register” definiert.

III.4.5 Service Data Objects (SDO)

Mit Hilfe der SDOs wird der Zugriff auf das Objektverzeichnis realisiert. Die SDOs werden zur Parametrierung und zur Statusabfrage verwendet. Der Zugriff auf ein einzelnes Objekt erfolgt mit einem Multiplexer über Index und Subindex des Objektverzeichnisses. Es werden die folgenden Kommunikationsprotokolle vom SERVOSTAR unterstützt:

- Initiate SDO Download Protocol
- Download SDO Segment Protocol
- Initiate SDO Upload Protocol
- Upload SDO Segment Protocol
- Abort SDO Transfer Protocol

Die Definition der einzelnen Kommunikationsdienste und der Protokolle sind dem DS301 zu entnehmen. Beispiele zur Handhabung von SDOs sind im Applikationsteil dieses Handbuches zu finden.



Achtung!
Da es sich bei einem SDO um einen bestätigten Dienst handelt, muss immer auf das SDO - Antworttelegramm gewartet werden, bevor ein neues Telegramm gesendet werden darf.

III.4.5.1 Aufbau eines Service Data Object

Ein SDO setzt sich aus folgenden Komponenten zusammen:

Byte	1	2	3	4	5	6	7	8
Inhalt	r/w	Index		Subindex	Daten			

1. Das Kontrollbyte (Byte 1):
Mit Hilfe des Kontrollbytes wird bestimmt, ob über das SDO schreibend oder lesend auf den Objektverzeichniseintrag zugegriffen wird. Eine Darstellung des gesamten Objektverzeichnis des SERVOSTAR finden Sie in Kapitel VI.3.
Der Datenaustausch mit dem SERVOSTAR hält sich an den Standard “CMS multiplexed domain protocols”, wie er im CAN-Standard DS 202 beschrieben wird.
Um Daten zu lesen, muss das Kontrollbyte entsprechend folgender Darstellung beschrieben werden:

Bit	7	6	5	4	3	2	1	0
Inhalt	ccs*=2			X	X	X	X	X

* ccs ⇒ client command specifier (ccs = 2 ⇒ initiate download request)
X ⇒ beliebig

Bei vier Byte im Nutzdatenfeld wird also ein Wert im Kontrollbyte von 0010 0011 binär bzw. 23_h übertragen.

Der Servoverstärker antwortet entsprechend mit einem Antwortbyte:

Bit	7	6	5	4	3	2	1	0
Inhalt	scs*=2			X				

* scs ⇒ server command specifier (scs = 2 ⇒ initiate download response)
n ⇒ nur gültig bei e = s = 1, Ist dies der Fall, so steht in n die Zahl der Bytes, die keine Daten enthalten.
X ⇒ beliebig

Bei erfolgreichem Lesen antwortet der Verstärker mit scs = 2, die Zahl der gültigen Bytes ist über n definiert, bei einem Fehler z.B. im Indexfeld k mit scs=4. Die Aufschlüsselung des Fehlers ist in Kapitel III.4.5.6 zu finden.
Um Daten zu schreiben, muss das Kontrollbyte entsprechend folgender Darstellung beschrieben werden:

Client	*Initiate Domain Download*															Server
	Byte 1								**2**	**3**	**4**	**5**	**6**	**7**	**8**	
request	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**								indication
⇒	ccs=1			X	n		e	s	m			d				⇒

⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒⇒

⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐⇐

	Byte 1								**2**	**3**	**4**	**5**	**6**	**7**	**8**	
⇐																⇐
confirm	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**								response
	scs=3			X					m			reserved				

2. Der Index (Bytes 2 und 3):
   Der Index ist der Haupteintrag im Objektverzeichnis und teilt die Parameter in Gruppen ein (Beispiel: Index 2022_h – Positionierdaten für den Mode Lage). Wie bei allen CAN – Daten wird der Index in umgekehrter Reihenfolge in den Bytes abgelegt, z.B.: Index 6040_h -> Byte 2 = 40_h, Byte 3 = 60_h)
3. Der Subindex (Byte 4):
   Der Subindex unterteilt die Parameter innerhalb einer Parametergruppe.
4. Datenfeld (Bytes 5 – 8):
   Über diese Komponenten werden die eigentlichen Nutzdaten ausgetauscht. In den Telegrammen zur Leseaufforderung an den SERVOSTAR sind diese auf 0 zu setzen, in der Schreibbestätigung vom SERVOSTAR haben diese Daten bei erfolgreichem Transfer keinen Inhalt, bei fehlerhaftem Schreiben enthalten sie einen Fehlercode (⇒ III.4.5.6).

III.4.5.2 Initiate SDO Download Protocol

Das “Initiate SDO Download” – Protokoll wird für Schreibzugriffe auf Objekte mit bis zu 4 Byte Nutzdaten verwendet (“expedited transfer”) oder zum Einleiten eines Segmenttransfers (“normal transfer”).

Der “expedited transfer” wird auch für Objekte genutzt, die nur eine Kommandoeigenschaft aufweisen (z.B. ASCII: SAVE) und somit keine weiteren Nutzdaten benötigen.

III.4.5.3 Download SDO Segment Protocol

Das “Download SDO Segment” – Protokoll wird für Schreibzugriffe auf Objekte mit mehr als 4 Byte Nutzdaten verwendet (“normal transfer”). Dieser Dienst wird vom SERVOSTAR z.Zt. nicht unterstützt, da keine Objekte existieren, die mehr als 4 Byte Nutzdaten verarbeiten.

III.4.5.4 Initiate SDO Upload Protocol

Das “Initiate SDO Upload” – Protokoll wird für Lesezugriffe auf Objekte mit bis zu 4 Byte Nutzdaten (“expedited transfer”) oder zum Einleiten eines Segmenttransfers (“normal transfer”) verwendet .

III.4.5.5 Upload SDO Segment Protocol

Das “Upload SDO Segment” – Protokoll wird für Lesezugriffe auf Objekte mit mehr als 4 Byte Nutzdaten verwendet (“normal transfer”). Dieser Dienst wird vom SERVOSTAR z.Zt. nicht unterstützt, da keine Objekte existieren, die mehr als 4 Byte Nutzdaten verarbeiten.

III.4.5.6 Abort SDO Protocol

Das "Abort SDO" Protokoll wird bei einer fehlerhaften SDO-Übertragung vom Servostar verwendet und zeigt mit Hilfe des "abort code" (Fehlercode) den Fehler an, der zum Abbruch der Übertragung führte. Der Fehlercode ist als UNSIGNED32-Wert kodiert. Die folgende Tabelle zeigt die möglichen Ursachen für einen Abbruch:

Abort Code	Beschreibung
0503 0000_h	“Toggle Bit” wurde nicht alterniert
0504 0000_h	SDO Protokoll “time out”
0504 0001_h	Client/Server Kommando - Bezeichner nicht gültig oder unbekannt
0504 0002_h	Unbekannte Blockgröße (block mode only)
0504 0003_h	Unbekannte Blocknummer (block mode only)
0504 0004_h	CRC Fehler (block mode only)
0504 0005_h	Außerhalb des Speichers
0601 0000_h	Zugriff auf dieses Objekt wird nicht unterstützt
0601 0001_h	Versuchter Lesezugriff auf ein Objekt, das nur geschrieben werden kann
0601 0002_h	Versuchter Schreibzugriff auf ein Objekt, das nur gelesen werden kann
0602 0000_h	Objekt existiert im Objektverzeichnis nicht
0604 0041_h	Objekt kann nicht in ein PDO “gemappt” werden
0604 0042_h	Größe und Anzahl der “gemappten” Objekte übersteigt die mögliche PDO Länge
0604 0043_h	Allgemeine Parameter - Inkompatibilität
0604 0047_h	Allgemeine Inkompatibilität im Gerät
0606 0000_h	Zugriffsverletzung aufgrund eines Hardwarefehlers
0607 0010_h	Datentyp passt nicht, Länge des Service Parameters passt nicht
0607 0012_h	Datentyp passt nicht, Länge des Service Parameters zu groß
0607 0013_h	Datentyp passt nicht, Länge des Service Parameters zu klein
0609 0011_h	Subindex existiert nicht
0609 0030_h	Wertebereich des Parameters verlassen (nur für Schreibzugriffe)
0609 0031_h	Wert des Parameters zu groß
0609 0032_h	Wert des Parameters zu klein
0609 0036_h	Maximaler Wert ist kleiner als minimaler Wert
0800 0000_h	Allgemeiner Fehler
0800 0020_h	Datum kann nicht übertragen oder gespeichert werden
0800 0021_h	Datum kann aufgrund lokaler Geräteführung nicht übertragen / gespeichert werden
0800 0022_h	Datum kann aufgrund des Gerätezustandes nicht übertragen / gespeichert werden
0800 0023_h	Dynamisches Generieren des Objektverzeichnisses nicht möglich oder bereits vorhanden (Objektverzeichnis wird aus einer Datei erstellt und ein Fehler aufgrund einer defekten Datei tritt auf)

Nicht aufgeführte “Abort Codes” sind reserviert.

III.4.6 Process Data Object (PDO)

PDOs werden für die Echtzeit - Datenkommunikation verwendet. Zum Beispiel lassen sich über PDOs ähnliche Regler aufbauen wie sie bei analog realisierten Antrieben verwendet werden. Statt der dort verwendeten +/- 10V Sollwertvorgabe und ROD-Istwertausgabe können hier digitaler Drehzahlsollwert und Istpositionsausgabe über PDOs realisiert werden. Die PDO-Daten werden dazu ohne Protokoll-Overhead übertragen und der Empfang nicht bestätigt.
Dieses Kommunikationsobjekt nutzt einen unbestätigten Kommunikationsdienst.
PDOs definieren sich über das Objektverzeichnis des SERVOSTAR. Hierbei können vordefinierte PDOs ausgewählt (“mapping” vordefinierter PDOs) oder vom Anwender selbst zusammengestellt (variables “mapping”) werden. Das “Mapping” erfolgt mit Hilfe von SDOs während der Konfigurationsphase. Die Länge und die “Mapping” Nummer des PDOs werden über das Antriebsprofil DS 402 definiert.

Die Definition des PDO Dienstes und des Protokolls sind dem DS301 zu entnehmen. Beispiele zur Handhabung von PDOs sind im Applikationsteil dieser Dokumentation zu finden.
Grundsätzlich können zwei Arten von PDOs in Abhängigkeit der Übertragungsrichtung unterschieden werden:

- Transmit PDOs (TPDOs) (SERVOSTAR ⇒ Master)
  Die TPDOs übertragen Daten vom SERVOSTAR zur Steuerung
  (z.B. Istwertobjekte, Gerätezustände).
- Receive PDOs (RPDOs) (Master ⇒ SERVOSTAR)
  Die RPDOs übertragen Daten von der Steuerung zum SERVOSTAR
  (z.B. Sollwerte)

Der SERVOSTAR unterstützt für jede Übertragungsrichtung zwei unabhängige PDO - Kanäle. Die Kanäle sind mit den Kanalnummern 1 bis 4 gekennzeichnet.
Zur Konfiguration der vier möglichen PDOs stehen jeweils drei Parameter zur Verfügung, die über entsprechende SDOs eingestellt werden können:

1. Auswahlparameter, mit denen aus einer Anzahl möglicher PDO-Konfigurationen die zwei PDOs in jeder Richtung ausgewählt werden, die im Prozessdatenbetrieb verwendet werden sollen:
   (SDOs 2600_h bis 2603_h, $2A00_h$ bis $2A03_h$)
2. Mappingparameter, über die zum einen ermittelt werden kann, welche Daten im ausgewählten PDO vorhanden (= gemappt) sind und über die zum anderen bei den frei konfigurierbaren PDOs (PDO. Nr. 37 bis 40) das Mapping der PDOs vorgegeben werden kann
3. Kommunikationsparameter, die festlegen, ob die PDOs im synchronisierten Betrieb oder ereignisgesteuert arbeiten sollen: (SDOs 1400_h bis 1403_h, 1800_h bis 1803_h)

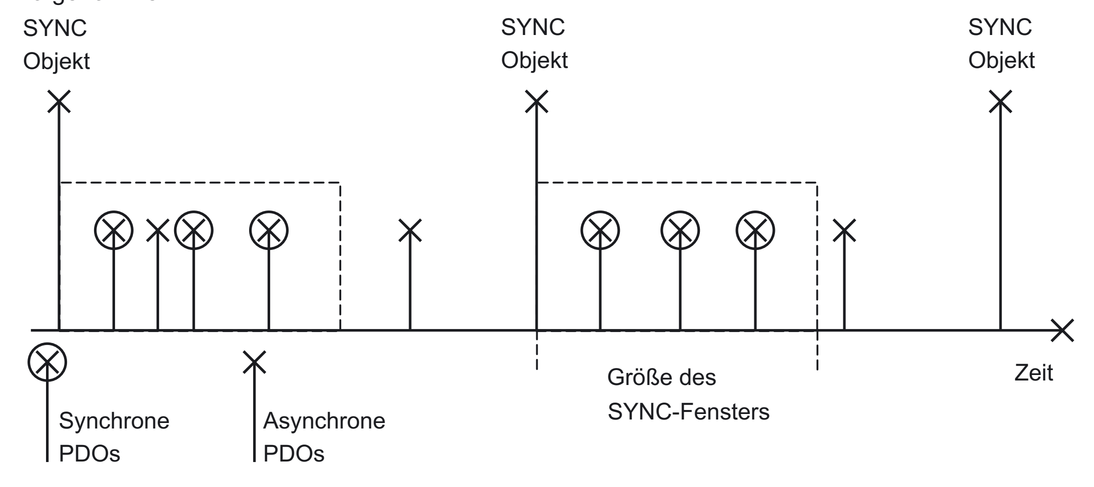
Darüber hinaus können einzelne Bits der TPDOs maskiert werden, um so bitweise das selbständige Generieren des Sendetriggers auf einzelne “Bitereignisse” von TPDOs zu steuern.

III.4.6.1 Übertragungsmodi

Es werden die folgenden PDO Übertragungsmodi (transmission mode) unterschieden:

- Synchrone Übertragung
- Asynchrone Übertragung

Um Antriebe zu synchronisieren, wird periodisch das vordefinierte SYNC – Objekt übertragen (Bustakt). Synchrone PDOs werden innerhalb eines vordefinierten Zeitfensters im Anschluss an das SYNC Objekt übertragen.
Die Einstellung der Übertragungsmodi wird mit Hilfe der PDO Kommunikationsparameter vorgenommen.



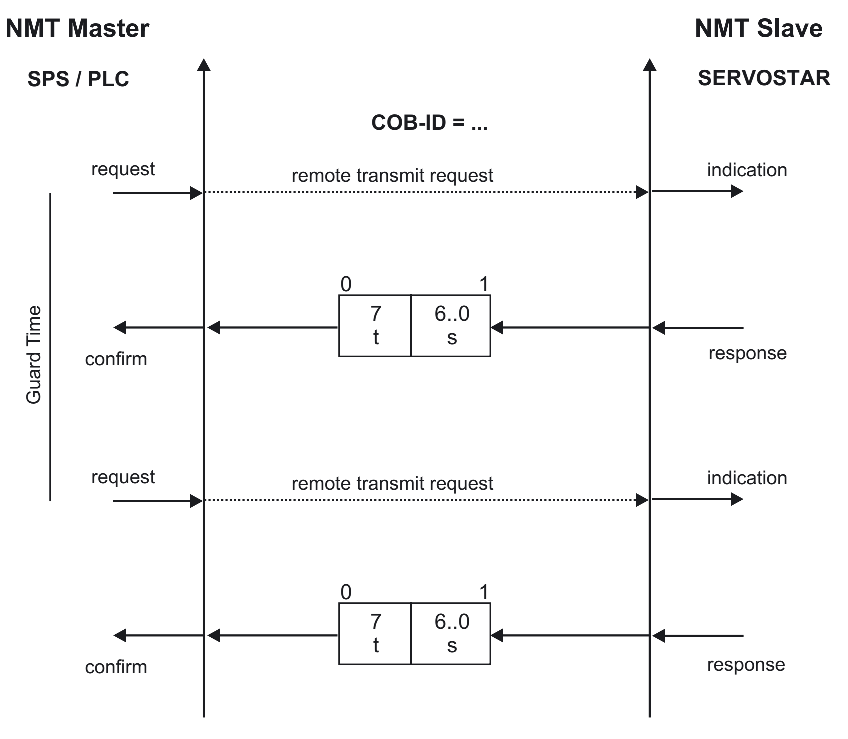
III.4.6.2 Triggermodi

Es werden drei Triggermodi (Triggering Modes) unterschieden:

- **Ereignisgesteuert (Event Driven)**
  Die Übertragung der Telegramme wird über ein objektspezifisches Ereignis getriggert. Zusätzlich besteht die Möglichkeit einzelne Bits objektunabhängig zu maskieren, um so das Selbständige generieren von Telegrammen einzuschränken und somit die Buslast zu senken (⇒ IV.3.2.2.4 f).
- **Zeitgesteuert (Time Driven)**
  Bei ereignisgesteuerten Signalen, die eine zu hohe Buslast erzeugen würden, kann mit Hilfe der "inhibit time" (Kommunikationsparameter, Subindex 3) betimmt werden, nach welcher Mindestzeit wieder ein PDO gesendet werden darf.
- **“Remote” Abfrage (Remotely requested)**
  Ein TPDO kann mit Hilfe eines “Remote” Telegramms angefordert werden.

III.4.7 Nodeguard

Das Node Guarding Protokoll dient der Funktionsüberwachung des Antriebs. Dazu wird der Antrieb in äquidistanten Zeitabständen vom CANopen - Master angesprochen.
Der maximale zeitliche Abstand, der zwischen zwei Nodeguard - Telegrammen liegen darf, wird durch das Produkt der Guard Time (SDO $100C_h$, ⇒ IV.2.1.12) und des Life Time Factors (SDO $100D_h$, ⇒ IV.2.1.13) bestimmt. Ist einer dieser beiden Werte 0, ist die Ansprechüberwachung deaktiviert. Das Nodeguarding wird nur bei enableter Endstufe aktiviert. Wird der Antrieb innerhalb der Zeit, die durch die SDOs $100C_h$ und $100D_h$ definiert ist, nicht angesprochen, tritt die Warnung N04 (Ansprechüberwachung) am Antrieb auf, der Antrieb wird mittels Quickstop - Rampe abgebremst und jede weitere Bewegung wird verhindert. Die zeitliche Folge des Nodeguardings sieht wie folgt aus:



t = toggle Bit, ändert seinen Zustand mit jedem Slave-Telegramm
s = Zustand der NMT Slave - Zustandsmaschine
Das Nodeguarding wird vom Master mit RTR - Telegrammen mit dem COB-ID ausgeführt, der über das SDO $100E_h$ (⇒ IV.2.1.14) beim Slave eingestellt werden kann. Der Defaultwert hierfür ist 700_h + Slave - Knotenadresse.

Diese Seite wurde bewusst leer gelassen.

IV CANopen Antriebsprofil

IV.1 Emergency Messages

“Emergency Messages” werden durch interne Gerätefehler ausgelöst. Sie haben eine hohe ID – Priorität, um einen schnellen Buszugriff sicherzustellen. Die “Emergency Message” beinhaltet ein Fehlerfeld mit vordefinierten Fehlernummern (2 Byte), Fehlerregister (1Byte) und die Fehlerkategorie (1Byte) und zusätzlichen Informationen (⇒ Kap. III). Das höherwertige Byte der Fehlernummer gibt die Fehlerklasse und das niederwertige Byte die Fehlernummer der Klasse an. Fehlernummern von $xx00_h$ bis $xx7F_h$ sind im Kommunikations- oder Antriebsprofil definiert. Fehlernummern von $xx80_h$ bis $xxFF_h$ sind herstellerspezifische Definitionen. Mit Hilfe der Fehlerkategorie können auftretende Fehler in ihrer Bedeutung klassifiziert werden. Es sind die folgenden Fehlerkategorien definiert:

1: Fehler, die nur durch einen Reset (Kommando “COLDSTART” oder Bit 7 im controlword ⇒ IV.4.2.1) gelöscht werden können. Diese Fehler werden auch durch Blinken der LED - Anzeige in der Frontplatte angezeigt.(Fxx, xx = Fehlernummer)

2: Fehler, die durch Bit 11 im controlword (⇒ Kap. IV.4.2.1) gelöscht werden können.

3: Fehlermeldungen, die bei der Verarbeitung eines PDOs auftreten können.

4: Fehler, die **nicht** durch den Anwender behoben werden können.

5: Bedienungsfehler / Warnungen

Die folgende Tabelle beschreibt die verschiedenen Error Codes:

Error Code	Kategorie	Beschreibung
0000$_h$	—	Error reset or no error (mandatory)
1000$_h$	—	Generic error (mandatory)
1080$_h$	5	Kein BTB (Zustand “not ready for operation”)
2330$_h$	2	Erdschluß (F22)
3100$_h$	1	Kein Netz-BTB (F16)
3110$_h$	1	Überspannung Zwischenkreis (F02)
3120$_h$	1	Unterspannung Zwischenkreis (F05)
3130$_h$	1	Eine Netzphase fehlt (bei PMODE = 2) (F19)
4110$_h$	1	Umgebungstemperatur überschritten (F13)
4210$_h$	1	Kühlkörpertemperatur überschritten (F01)
4310$_h$	1	Motortemperatur überschritten (F06)
5111$_h$	1	Fehler ±15VHilfsspannung (F07)
5380$_h$	1	Fehler A/D-Konverter (F17)
5400$_h$	1	Endstufenfehler (F14)
5420$_h$	1	Ballast (Chopper) (F18)
5441$_h$	1	Fehler bei Bedienung der AS - Option (F27)
5530$_h$	1	Serielles EEPROM (F09)
5581$_h$	1	Flash-EEPROM (F10)
6010$_h$	4	Watchdog (Softwarereset, F32)
6181$_h$	4	BCC Fehler (Table)
6182$_h$	4	BCC Fehler (System Makro)
6183$_h$	4	BCC Fehler (EEPROM seriell)
6184$_h$	4	FPGA Fehler
6185$_h$	4	Fehler (Table)
6281$_h$	4	User Software BCC (Makro, F32)
6282$_h$	4	Fehlerhafte User Software (Makro, F32)
6320$_h$	3	Parameterfehler
7111$_h$	1	Bremsenfehler (F11)
7122$_h$	1	Kommutierungsfehler (F25)
7181$_h$	5	SERVOSTAR konnte nicht enabled werden
7182$_h$	5	Kommando nur im Zustand “disabled” möglich
7303$_h$	1	Fehler Rückführeinheit (F04)
8053$_h$	1	Handlingfehler (F21)
8181$_h$	2	Ansprechüberwachung aktiv
8182$_h$	1	CAN - Busoff (F23)
8281$_h$	5	Zustandsmaschine ist nicht im “Operational Enable”-Zustand
8282$_h$	5	Fehlerhafte Modeeinstellung
8331$_h$	1	I^{2}t (torque fault, F15)
8480$_h$	1	Überdrehzahl (overspeed, F08)
8611$_h$	2	Schleppfehler
8681$_h$	5	Ungültige Fahrauftragsnummer
8682$_h$	2	Fehler externe Trajektorie (F28) (nur Sercos)
FF01$_h$	4	Schwerwiegender Ausnahmefehler (F32)
FF02$_h$	3	Fehler in PDO-Komponente
FF03$_h$	5	Betriebsmode
FF04	1	Slotfehler (F20)
FF06	2	Warnungsanzeige als Fehler (F24)
FF07	2	Fehler Referenzfahrt (HW-Endschalter erreicht) (F26)
FF08	2	Sercos - Fehler (F29)
FF11	1	Sercos

IV.2 Allgemeine Definationen

In diesem Kapitel werden allgemeingültige Objekte beschrieben (z.B. SDO 1000_h “Device Type”). Im Anschluss daran wird die freie Konfiguration von Prozessdatenobjekten (“free mapping”) erläutert.

IV.2.1 Allgemeine Objekte

IV.2.1.1 SDO 1000h: Device Type (DS301)

Dieses Objekt beschreibt den Gerätetyp (Servoantrieb) und die Gerätefunktionalität (DS402 Antriebsprofil). Es setzt sich folgendermaßen zusammen:

MSB LSB

zusätzliche Informationen		Geräte-Profil-Nummer
Modebits	Typ	402_d
31 24	23 16	15 0

Nummer des Geräteprofils: 402_d (Antriebsprofil)
Typ: 02 (Servoantrieb)
Modebits: 0 (Herstellerspezifisch)

Objektbeschreibung:

Index	1000_h
Name	Gerätetyp
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	M

Wertbeschreibung:

Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	nein

IV.2.1.2 SDO 1001h: Error register (DS301)

Wird ein Fehlerbit im herstellerunabhängigen Fehlerregister gesetzt, so werden detailliertere Informationen im SDO 1003_h zur Verfügung gestellt. Dieses Objekt ist Teil des Fehlerobjekts (Emergency Message).

Objektbeschreibung:

Index	1001_h
Name	Error register
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	M

Wertbeschreibung:

Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	nein

Im Folgenden wird die Bitbelegung des Fehlerregisters beschrieben. Ein gesetztes Bit zeigt ein Fehlerereignis an.

Bit	Beschreibung	Bit	Beschreibung
0	generic error	4	communication error
1	current	5	device profile specific
2	voltage	6	reserved
3	temperature	7	manufacturer specific

IV.2.1.3 SDO 1002h: Manufacturer Status Register (DS301)

Dieses Objekt beinhaltet das herstellerspezifische Statusregister, in dem auch die SERVOSTAR Warnungen enthalten sind (s. auch ASCII Objekt DRVSTAT).

Objektbeschreibung:

Index	1002_h
Name	Manufacturer Status Register
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	M

Wertbeschreibung:

Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED8
Defaultwert	nein

Die folgende Tabelle zeigt die Bitbelegung des Statusregisters:

Bit	Beschreibung
0	Warnung 1: I²t-Meldeschwelle überschritten
1	Warnung 2: Ballastleistung erreicht
2	Warnung 3: Schleppfehler
3	Warnung 4: Ansprechüberwachung aktiv
4	Warnung 5: Netzphase fehlt
5	Warnung 6: Software-Endschalter 1 hat angesprochen
6	Warnung 7: Software-Endschalter 2 hat angesprochen
7	Warnung 8: Fehlerhafter Fahrauftrag gestartet
8	Warnung 9: Beim Fahrauftrag-Start war kein Referenzpunkt gesetzt
9	Warnung 10: PSTOP aktiv
10	Warnung 11: NSTOP aktiv
11	Warnung 12: Motordefaultwerte wurden geladen
12	Warnung 13: Erweiterungskarte arbeitet nicht ordnungsgemäß
13	Warnung 14: Motorphase
14	Warnung 15: fehlerhafter VCT-Eintrag angewählt
15	Warnung 16: Reserve
16	Fahrauftrag aktiv
17	Referenzpunkt gesetzt
18	Aktuelle Position = Home Position
19	In Position
20	Position-Latch erfolgt (positive Flanke)
21	Position 0 erreicht
22	Position 1 erreicht
23	Position 2 erreicht
24	Position 3 erreicht
25	Position 4 erreicht
26	Initialisierung beendet
27	Position 5 erreicht
28	Drehzahl = 0
29	Sicherheitsrelais hat angesprochen
30	Endstufe freigegeben
31	Fehler steht an

IV.2.1.4 SDO 1003h: Pre-Defined Error Field (DS301)

Dieses Objekt beinhaltet das letzte Fehlerereignis, das über ein Fehlertelegramm gemeldet wurde. Es wird nur der letzte Fehler angezeigt.

1. Der Wert unter Subindex 00_h zeigt die Anzahl der aufgezeichneten Fehler an.
2. Der aktuellste Fehler steht unter Subindex 01_h.
3. Wird der Wert 0 in Subindex 02_h geschrieben, wird der Fehler aus der Fehlerliste gelöscht (Achtung: Fehler wird nicht quittiert!).
4. Die Fehlernummer ist vom Datentyp UNSIGNED32 und setzt sich aus einer 16 Bit Fehlernummer und einem herstellerspezifischen Informationsfeld zusammen.
   Das herstellerspezifische Informationsfeld wird vom SERVOSTAR nicht benutzt.
   Die möglichen Fehlernummern sind im Kapitel “Error Message” (⇒ Kap. IV.1) beschrieben.
   Das Datenfeld baut sich folgendermaßen auf:

MSB	**LSB**
Informationsfeld	Fehlernummer

Objektbeschreibung:

Index	1003_h
Name	Error Field
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Subindex	00_h
Beschreibung	Anzahl der Fehler
Kategorie	M
Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	0..254
Defaultwert	nein

Subindex	01_h
Beschreibung	Standard Fehlerfeld (⇒ Kap. IV.1)
Kategorie	M
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	nein

IV.2.1.5 SDO 1004h: Zahl der unterstützten PDOs (DS301)

Dieses Objekt wird nur aus Kompatibilitätsgründen zu älteren Ausgaben des CANopen-Standards DS 301 unterstützt.

Parameterstruktur:

Subindex	MSB	LSB
0	Zahl der unterstützten RPDOs	Zahl der unterstützten TPDOs
1	Zahl der unterstützten synchronen RPDOs	Zahl der unterstützten synchronen TPDOs
2	Zahl der unterstützten asynchronen RPDOs	Zahl der unterstützten asynchronen TPDOs

Objektbeschreibung:

Index	1004_h
Name	Zahl der unterstützten PDOs
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Subindex	00_h
Beschreibung	Zahl der unterstützten PDOs
Kategorie	O
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	00_h - $1FF01FF_h$
Defaultwert	00020002_h

Subindex	01_h
Beschreibung	Zahl der synchronen PDOs
Kategorie	O
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	00_h - $1FF01FF_h$
Defaultwert	00_h

Subindex	02_h
Beschreibung	Zahl der asynchronen PDOs
Kategorie	O
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	00_h - $1FF01FF_h$
Defaultwert	00020002_h

Beschrieben ist hier nur die Verteilung nach dem Hochlauf des SERVOSTAR. Danach können die PDOs über die Kommunikationsparameter 1400_h bis 1403_h, 1800_h bis 1803_h konfiguriert werden.

IV.2.1.6 SDO 1005h: COB-ID der SYNC - Message (DS301)

Mit diesem Objekt kann der COB - ID für die SYNC - Message geändert werden.

Objektbeschreibung:

Index	1005_h
Name	COB - ID der SYNC message
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	80000080_h

Parameterstruktur:

MSB				LSB
X	0/1	0	00_h	11 Bit identifier
31	30	29	28 ... 11	10 ... 0

Nur die Bits 0 - 10, 29 und 30 haben dabei eine Bedeutung:

Bit	Wert	Bedeutung
31	X	—
30	0	Gerät erzeugt keine SYNC-Message
	1	Gerät erzeugt SYNC-Message
29	0	11 Bit ID
	1	29 Bit ID
28..11	X	Bit 11..28 von 29 Bit SYNC COB-ID
10..0	X	Bit 0..10 von SYNC COB-ID

IV.2.1.7 SDO 1006h: Communication Cycle Period (DS301)

Mit diesem Objekt kann die Periode des Sendens des SYNC - Telegramms in µs festgelegt werden.

Objektbeschreibung:

Index	1006_h
Name	Periode des Kommunikationszyklus
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	00_h

IV.2.1.8 SDO 1007h: Synchronous window length (DS301)

Dieses Objekt legt die Länge des Zeitfensters für synchrone PDOs in µs fest.

Objektbeschreibung:

Index	1007_h
Name	Zeitfenster für synchrone Nachrichten
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	00_h

IV.2.1.9 SDO 1008h: Manufacturer Device Name (DS301)

Der Gerätename besteht aus vier ASCII-Zeichen und hat die Buchstaben “SDxx” zum Inhalt, wobei xx für die Stromstärke der Endstufe steht (z.B. SD06).

Objektbeschreibung:

Index	1008_h
Name	Manufacturer Device Name
Objektcode	VAR
Datentyp	Visible String
Kategorie	Optional

Wertbeschreibung:

Zugriff	const
PDO Mapping	nicht möglich
Wertebereich	nein
Defaultwert	nein

IV.2.1.10 SDO 100Ah: Manufacturer Software Version (DS301)

Die Interface-Softwareversion besteht aus vier ASCII-Zeichen (z.B. 3.00).

Objektbeschreibung:

Index	$100A_h$
Name	Manufacturer Software Version
Objektcode	VAR
Datentyp	Visible String
Kategorie	Optional

Wertbeschreibung:

Zugriff	const
PDO Mapping	nicht möglich
Wertebereich	nein
Defaultwert	nein

IV.2.1.11 SDO 100Bh: Node-ID (DS301)

Über das Objekt “Node-ID” kann die Teilnehmeradresse des SERVOSTAR ausgelesen werden. Die Teilnehmeradresse kann unter anderem über das ASCII-Objekt "ADDR" verändert werden. Der einzustellende Adressbereich (Wertebereich) ist abhängig vom ASCII – Objekt “MDRV”. Die Teilnehmeradresse kann von 1..63 oder von 1..127 eingestellt werden.

Wird der expandierte Adressbereich verwendet, kann die Inbetriebnahmesoftware nicht mehr über den CAN – Bus vernetzt werden.

Objektbeschreibung:

Index	$100B_h$
Name	Node-ID
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	Optional

Wertbeschreibung:

Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	1..127 (1..63)
Defaultwert	1

IV.2.1.12 SDO 100Ch: Guard Time (DS301)

Das Produkt der SDOs $100C_h$ "Guard Time“ und $100D_h$ "Life Time Factor“ ergibt die Ansprechüberwachungszeit. Die “Guard Time“ wird in Millisekunden angegeben. Die Ansprechüberwachung wird erst mit dem ersten “Nodeguard - Objekt“ aktiv (⇒ Kap. III.4.7). Wird der Wert des Objekts “Guard Time“ zu Null gesetzt, ist die Ansprechüberwachung inaktiv.

Objektbeschreibung:

Index	$100C_h$
Name	Guard Time
Objektcode	VAR
Datentyp	UNSIGNED16
Kategorie	Mandatory

Wertbeschreibung:

Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED16
Defaultwert	0

IV.2.1.13 SDO 100Dh: Life Time Factor (DS301)

Das Produkt der Objekte “Guard Time“ und “Life Time Factor“ ergibt die Ansprechüberwachungszeit. Die Ansprechüberwachung wird erst mit dem ersten “Nodeguard - Objekt“ aktiv (⇒ III.4.7). Wird der Wert des Objekts “Life Time Factor“ zu Null gesetzt, ist die Ansprechüberwachung inaktiv.

Objektbeschreibung:

Index	$100D_h$
Name	Life Time Factor
Objektcode	VAR
Datentyp	UNSIGNED8
Kategorie	Mandatory

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	0..255
Defaultwert	0

IV.2.1.14 SDO 100Eh: COB - ID Nodeguarding (DS301)

Dieses Objekt legt den COB - ID für das Nodeguarding - Protokoll fest.

Objektbeschreibung:

Index	$100E_h$
Name	Node guarding identifier
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	700 + node - ID

IV.2.1.15 SDO 100Fh: Zahl der unterstützten SDOs (DS301)

Dieses Objekt beschreibt die Zahl der vom Gerät unterstützten SDOs.

Objektbeschreibung:

Index	$100F_h$
Name	Zahl der unterstützten SDOs
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	01_h

IV.2.1.16 SDO 1010h: Store Parameters (DS301)

Mit Hilfe dieses Objekts können die Kommunikations- und Mappingparameter abgespeichert werden.

Objektbeschreibung:

Index	1010_h
Name	store parameters
Objektcode	ARRAY
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Subindex	0_h
Beschreibung	Anzahl der Einträge
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	2
Defaultwert	2

Subindex	1_h
Beschreibung	save all parameters
Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	0

Dieser Subindex ist reserviert.

Subindex	2_h
Beschreibung	save communication parameters
Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	0

Zum Abspeichern der Kommunikations- und Mappingparameter muss Subindex 2 verwendet werden. Es muss ein schreibender Zugriff mit der Schreibsignatur "save" erfolgen, d.h. Sie müssen folgenden Wert schreiben:

MSB			LSB
e	v	a	s
65_h	76_h	61_h	73_h

Ein Abspeichern der Parameter kann nur erfolgen, wenn diese Funktion über den Parameter DRVCNFG (SDO 6372, Subindex 1) Bit 3 freigegeben ist.

IV.2.1.17 SDO 1012h: COB - ID für Time - Stamp Message (DS301)

Mit diesem Objekt kann der COB - ID für die Time - stamp - Message definiert werden.

Objektbeschreibung:

Index	1012_h
Name	COB-ID time stamp message
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	100_h

IV.2.1.18 SDO 1013h: High Resolution Time Stamp (i.V.) (DS301)

Mit diesem Objekt kann ein Zeitwert gelesen oder geschrieben werden.

Objektbeschreibung:

Index	1013_h
Name	high resolution time stamp
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	00_h

IV.2.1.19 SDO 1014h: COB - ID für Emergency Message (DS301)

Mit diesem Objekt kann der COB - ID für die Emergency definiert werden.

Objektbeschreibung:

Index	1014_h
Name	COB-ID emergency message
Objektcode	VAR
Datentyp	UNSIGNED32
Kategorie	O

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	80_h + Node - ID

IV.2.1.20 SDO 1018h: Identity Object (DS301)

Das “Identity Objekt” beinhaltet allgemeine Geräteinformationen.

Objektbeschreibung:

Index	1018_h
Name	Identity Object
Objektcode	RECORD
Datentyp	Identity

Wertbeschreibung:

Subindex	00_h
Beschreibung	Anzahl der Einträge
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	1..4
Defaultwert	4

Subindex	01_h
Beschreibung	Vendor ID*
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	$6A_H$

* 00 00 00 $6A_h$

Subindex	02_h
Beschreibung	Product Code*
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	nein

* Stromstärke des Verstärkers

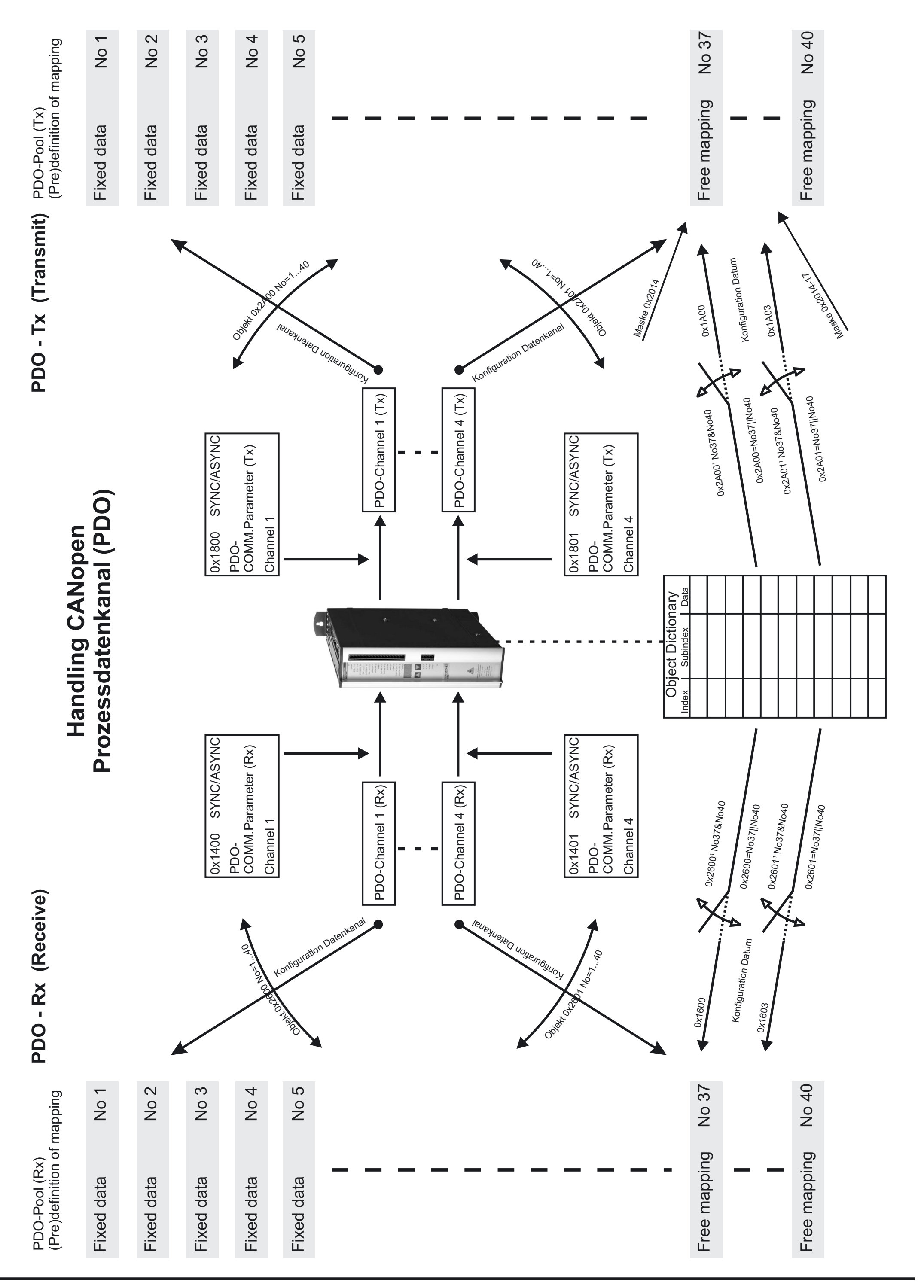
Subindex	03_h
Beschreibung	Revision Number*
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	nein

* Bit 16..31 Firmware Version, Bit 15..0 CANopen Version

Subindex	04_h
Beschreibung	Serial Number
Zugriff	r
PDO Mapping	nicht möglich
Wertebereich	UNSIGNED32
Defaultwert	nein

IV.3 PDO Mapping

Da der SERVOSTAR mehr als einen Betriebsmode unterstützt, werden unterschiedlich konfigurierte PDOs für die Sende- und Empfangsrichtung - in Abhängigkeit der Anwendung - benötigt. Die erste Lösung für die Handhabung von frei konfigurierbaren PDOs im SERVOSTAR war das “quasi freie Mapping”. Bei dieser Art von freiem Mapping werden vordefinierte PDOs mit Hilfe eines Index aus der PDO Bibliothek ausgewählt. Diese PDOs können inhaltlich nicht mehr umkonfiguriert werden. Das “quasi freie Mapping” hat sich jedoch als zu statisch für die wachsende Anzahl unterschiedlichster Anwendungen gezeigt. Daher wurde das “vollständig freie Mapping” für PDOs zusätzlich eingeführt. Hierbei kann der Anwender das PDO auch inhaltlich verändern (⇒ III.4.6). Die folgende Abbildung “Handling CANopen Prozessdatenkanal” veranschaulicht die Handhabung von PDOs.



IV.3.1 Empfangs-PDOs (RPDO)

Mit den SDOs 2600_h bis 2603_h können RPDOs (R=receive) ausgewählt werden. Die Konfiguration der RPDOs erfolgt mit den SDOs 1400_h bis 1403_h (Kommunikationsparameter) und 1600_h bis 1603_h(Mapping - Parameter). Diese Objekte arbeiten kanalbezogen d. h.:

- RPDO Kanal 1: 2600_h, 1400_h, 1600_h
- RPDO Kanal 2: 2601_h, 1401_h, 1601_h
- RPDO Kanal 3: 2602_h, 1402_h, 1602_h
- RPDO Kanal 4: 2603_h, 1403_h, 1603_h

Die SDOs 1600_h bis 1603_h können nur gelesen und so die Objektkonfiguration innerhalb des RPDOs ermittelt werden.



Ausnahme:
Wurden einer oder mehrere der RPDOs 37 bis 40 ausgewählt, muss die Objektkonfiguration für den jeweiligen RPDO – Kanal mit den SDOs 1600_h bis 1603_h (Schreibzugriff) erfolgen.

Es sind die folgenden PDOs definiert:

PDO Nummer	Mapping Objekt Index	Subindex	Mapping Object Name
1	6040_h	—	Controlword
2..20	—	—	reserviert
21	3100_h	01_h..08_h	ASCII-Objekt
22	6040_h	—	controlword
	2060_h	00_h	Strom- oder Drehzahlsollwert
23..31	—	—	reserviert
32	2060_h	00_h	Strom- oder Drehzahlsollwert
33	2022_h	04_h	Inkrementeller Positionssollwert
	2022_h	04_h	Inkrementeller Positionssollwert
34	2022_h	01_h	Position
	2022_h	02_h	Geschwindigkeit
	2022_h	03_h	Fahrauftragsart
35	2022_h	05_h	Fahrsatznummer (Achtung: Im Gerätezustand “Operation Enable” wird sofort gestartet)
36	—	—	reserviert
37...40	$xxxx_h$	xx_h	Frei definierbare RPDOs
41..64	—	—	reserviert

IV.3.1.1 Beschreibung der vordefinierten Empfangs-PDOs

IV.3.1.1.1 PDO controlword (1) (DS402)

Das PDO controlword (Default - PDO) besteht aus dem controlword (UNSIGNED16). Mit diesem PDO kann am Einschalten ist dieses PDO auf RPDO 1 gemappt.
Die Tabelle beschreibt das Mapping des PDO controlword:

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
01_h	60400010_h	Kontrollwort

IV.3.1.1.2 PDO ASCII-Kanal empfangen (21)

Mit Hilfe des ASCII-Kanals (Default-PDO) können sämtliche Parameter und Kommandos zum SERVOSTAR übertragen werden. Es können bis zu 8 ASCII-Zeichen in einem PDO gesendet werden. Kommandos oder Parameter, die mehr als 8 Zeichen benötigen, müssen segmentiert werden. Alle Kommandos und Parameter werden mit dem ASCII-Code “CR LF” ($0xD_h$, $0xA_h$) abgeschlossen. Die nicht benötigten Bytes im PDO werden mit dem ASCII-Code “NUL” ($0x0_h$) aufgefüllt, da sonst jedes überflüssige Zeichen als neues Kommando erkannt würde. Nach dem Einschalten ist dieses PDO auf RPDO2 gemappt.
Die Tabelle beschreibt das Mapping des PDO ASCII-Kanal empfangen:

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
$01..08_h$	31000208_h	1. bis 8. ASCII-Zeichen

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

IV.3.1.1.3 PDO Strom- / Drehzahlsollwert (22)

Das PDO Strom- bzw. Drehzahlsollwert setzt sich aus dem controlword (UNSIGNED16) und dem Sollwert (Signed16) zusammen. Dieses PDO darf nur in den Modi “Drehzahl digital“ oder “Strom digital“ benutzt werden. Anhand des eingestellten Mode (digitaler Strom- bzw. Drehzahlsollwert) wird erkannt, ob es sich um einen Drehzahl- oder Stromsollwert handelt. Das PDO wird sofort ausgeführt. Mehrfaches Senden des PDO mit verschiedenen Sollwerten erfordert kein zwischenzeitliches Anhalten des Antriebs.

Stromnormierung: 3280 = Spitzenstrom des Reglers
1640 = Nennstrom

z.B. Nennstrom = 3A, Sollwert = 1,0A $\Rightarrow$ 547 Inkremente

Drehzahlnormierung:
z.B. Drehzahl = $3000 min^{-1}$ $\Rightarrow$ Sollwert = 419430

Die Tabelle beschreibt das Mapping des PDO Sollwert:

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
01_h	60400010_h	Controlword
02_h	20600020_h	Strom- / Drehzahlsollwert

IV.3.1.1.4 PDO Sollwert 2 (32)

Das PDO Sollwert_2 ist ein zeit- und datenoptimiertes PDO. Es beinhaltet ausschließlich einen 32 Bit Sollwert. Dieses PDO darf nur in den Modi “Drehzahl digital“ oder “Strom digital“ benutzt werden. Anhand des eingestellten Mode (digitaler Strom- bzw. Drehzahlsollwert) wird erkannt, ob es sich um einen Drehzahl- oder Stromsollwert handelt.
Das PDO wird sofort ausgeführt. Mehrfaches Senden des PDO mit verschiedenen Sollwerten erfordert kein zwischenzeitliches Anhalten des Antriebs.

Die Tabelle beschreibt das Mapping des PDO Sollwert2:

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
01_h	20600020_h	Strom- / Drehzahlsollwert

IV.3.1.1.5 PDO Trajektorie (33)

Das PDO Trajektorie ist ein zeit- und datenoptimiertes PDO. Dieses PDO darf nur im Mode “Trajektorie” benutzt werden. Das PDO Trajektorie muss in immer gleichen Zeitabständen (einstellbar mit dem Kommando “PTBASE”) übertragen werden, da es sonst zu Einbrüchen im Drehzahlverlauf kommt. Dieses PDO besteht aus zwei Komponenten, den inkrementellen Istwerten für zwei Achsen. Die Zuordnung der Daten für die Achsen, die beide auf den gleichen COB-ID für dieses RPDO eingestellt werden müssen (Subindex 01_h der entsprechenden Kommunikationsparameter), erfolgt über das SDO 2721_h.

Beispiel zur Berechnung der absoluten Position:

$$\text{Position} = \frac{\text{inkrementeller Positionswert}}{2^{20}}$$

Die max. Differenz zwischen zwei inkrementellen Positionen wird durch die eingestellte Enddrehzahl (ASCII - Kommando VLIM) vorgegeben (s. Beispiel).

Beispiel zur maximalen inkrementellen Positionsdifferenz:

$$\text{max.erreichbare Enddrehzahl} / 1000 \frac{\text{Umd.}}{\text{min}} = 0{,}016667 \frac{\text{Umd.}}{\text{ms}}$$

$$|\text{ink.Pos.}(t_2) - \text{ink.Pos.}(t_1)| \leq 2^{20} \times 0{,}016667 = 17475$$

In Abhängigkeit der eingestellten Verstärkerparameter kann es zu einem mehr oder weniger großen Schleppfehler kommen. Tritt die Fehlermeldung “Schleppfehler” auf und die Achse wird mit der Notbremsrampe angehalten, so kann das mehrere Fehlerursachen haben:

- Die inkrementelle Positionsdifferenz ist zu groß gewählt worden (s.o.).
- Das Schleppfehlerfenster ist zu klein eingestellt worden (SDO 2020_h Subindex 03_h)
- Verstärkerparameter sind nicht optimal eingestellt

Die Tabelle beschreibt das Mapping des PDO Trajektorie:

Subindex	Wert	Beschreibung
00_h	2_h	Anzahl der Einträge
01_h	20220420_h	Inkrementelle Position
02_h	20220420_h	Inkrementelle Position

Dieses Objekt unterstützt **nicht** den Übertragungstyp 255 (asynchron).

IV.3.1.1.6 PDO Fahrsatz (34)

Das PDO Fahrsatz setzt sich aus Position (SIGNED32, gewichtet), Geschwindigkeit (UNSIGNED16) und der Fahrauftragsart (UNSIGNED16) zusammen.
Das PDO startet einen Fahrsatz aus dem flüchtigen Fahrsatzspeicher (Fahrsatznummer = 0) und ist ausschließlich im Mode “Lage“ verwendbar.

Die Tabelle beschreibt das Mapping des PDO Fahrsatz:

Subindex	Wert	Beschreibung
00_h	3_h	Anzahl der Einträge
01_h	20220120_h	Position
02_h	20220210_h	Geschwindigkeit
03_h	20220310_h	Fahrauftragsart (abs. / rel.)

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

IV.3.1.1.7 PDO Starte Fahrsatz (35)

Das PDO Starte Fahrsatz besteht aus der Fahrauftragsnummer (UNSIGNED16).
Das PDO startet einen Fahrsatz aus dem flüchtigen (Fahrsatznummer = 0, 192..255) oder permanenten (Fahrsatznummer = 1..180) Fahrsatzspeicher.
Dieses PDO kann ausschließlich im Mode "Lage" verwendet werden.

Die Tabelle beschreibt das Mapping des PDO starte Fahrsatz:

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
01_h	20220510_h	Fahrauftragsnummer

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

IV.3.1.1.8 PDO Frei definierbar (37 bis 40)

Sind diese PDOs angewählt, so können die Objekte frei hinzugefügt werden. Hierzu werden die SDOs 1600_h bis 1603_h (Mapping Parameter) verwendet. Es können bis zu 8 einzelne Objekte in ein PDO “gemappt” werden.

Ein Beispiel zum PDO Mapping und Kommunikationskonfiguration finden Sie im Anhang
(⇒ Kap. VI.1).

IV.3.1.2 Objektbeschreibung

IV.3.1.2.1 SDO 1400-1403h: 1st -4th Receive - PDO Kommunikationparameter (DS301)

Subindex	Beschreibung
00_h	Anzahl der Einträge
01_h	PDO COB-ID*
02_h	transmission type**
03_h	inhibit time
04_h	CMS priority group

* Nach der Selektion eines PDOs über 2600-2603$_h$ kann hier der COB-Identifier benannt werden, auf den der Verstärker reagieren soll (falls ein abweichender Wert vom Defaultwert erforderlich ist)

** Zur Anwahl SYNC, event-gesteuert etc.

IV.3.1.2.2 SDO 1600-1603h: 1st -4th Receive - PDO Mappingparameter (DS301)

Subindex	Beschreibung
00_h	Anzahl der konfigurierten Objekte
$01_h..08_h$	Beschreibung der konfigurierten Objekte

IV.3.1.2.3 SDO 2600-2603h: 1st -4th Receive PDO select

Mit Hilfe dieses Objekts wird ein vordefiniertes Empfangs - PDO ausgewählt. Mit den SDOs 1400-1403$_h$ "1st -4th Receive PDO parameter" und 1600-1603$_h$ "1st -4th Receive PDO mapping" können dann die Eigenschaften dieses PDOs festgelegt werden.

Dieses Objekt ermöglicht ein variables Mapping von vordefinierten PDOs. Die möglichen anwählbaren PDOs sind in der Tabelle in Kapitel IV.3.1 aufgeführt.

Objektbeschreibung:

Index	2600_h ... 2603_h
Name	1st -4th Receive PDO select
Objektcode	VAR
Datentyp	UNSIGNED8

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	0..255
Defaultwert	0

IV.3.1.2.4 SDO 2721h: Konfiguration Receive PDO 33

Dieses Objekt beeinflusst die Verarbeitung der empfangenen acht Datenbytes des PDOs Trajektorie (Nr. 33). Mit dem Wert „LOW“ (LOW = 0) werden die Bytes 0 – 3 (Subindex 01_h des PDO-Mappings) und mit dem Wert „HIGH“ (HIGH = 1) die Bytes 4 – 7 (Subindex 02_h des PDO-Mappings) des PDOs als inkrementeller Sollwert für die nächste Bewegung interpretiert.

Objektbeschreibung:

Index	2721_h
Name	PDO 33 data select
Objektcode	VAR
Datentyp	UNSIGNED8

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	0, 4
EEPROM	nein
Defaultwert	0

IV.3.2 Sende-PDOs (TPDO)

Mit den SDOs $2A00_h$ bis $2A03_h$ können TPDOs (t=transmit) ausgewählt werden. Die Konfiguration der TPDOs erfolgt mit den SDOs 1800_h bis 1803_h (Kommunikationsparameter) und $1A00_h$ bis $1A03_h$ (Mapping - Parameter). Diese Objekte arbeiten kanalbezogen d. h.:

- TPDO Kanal 1: $2A00_h$, 1800_h, $1A00_h$
- TPDO Kanal 2: $2A01_h$, 1801_h, $1A01_h$
- TPDO Kanal 3: $2A02_h$, 1802_h, $1A02_h$
- TPDO Kanal 4: $2A03_h$, 1803_h, $1A03_h$

Die SDOs $1A00_h$ bis $1A03_h$ können nur gelesen werden und so die Objektkonfiguration innerhalb des PDOs ermittelt werden.



Ausnahme:
Wurden einer oder mehrere der TPDOs 37 bis 40 ausgewählt, muss die Objektkonfiguration für den jeweiligen TPDO – Kanal mit den SDOs $1A00_h$ bis $1A03_h$ (Schreibzugriff) erfolgen.

Es sind die folgenden PDOs definiert:

PDO Nummer	Mapping Objekt Index	Subindex	Mapping Object Name
1 (01_h)	6041_h	—	statusword
2..20 (02_h..14_h)	—		reserviert
21 (15_h)	3102_h	01_h..08_h	ASCII-Objekt
22 (16_h)	6041_h	—	statusword
	2070_h	01_h	Istlage (20 Bit / Umdrehung)
	2070_h	02_h	Drehzahl (1 Bit = 1875/262144 [Umdr./min])
23 (17_h)	6041_h	—	statusword
	1002_h	—	herstellerspezifisches Statusregister
24..31 (18_h..$1F_h$)	—	—	reserviert
32 (20_h)	2070_h	01_h	Istlage (20 Bit / Umdrehung)
	2070_h	02_h	Drehzahl (1 Bit = 1875/262144 [Umdr./min])
33 (21_h)	2070_h	03_h	Inkrementeller Positionsistwert
	2070_h	11_h	2. erw. Statusregister
34..35 (22_h..23_h)	—	—	—
36 (24_h)	—	—	reserviert
37...40 (25_h...28_h)	$xxxx_h$	xx_h	Frei definierbare TPDOs
41..64 (27_h..40_h)	—	—	reserviert

IV.3.2.1 Beschreibung der vordefinierten Sende-PDOs

IV.3.2.1.1 PDO statusword (1) (DS402)

Das PDO statusword (Default - PDO) besteht aus dem statusword (UNSIGNED16).
Mit diesem PDO kann ausschließlich der Zustand der Zustandsmaschine (⇒ IV.4.1) ermittelt werden. Dieses PDO ist modeunabhängig.
Nach dem Einschalten ist dieses PDO auf TPDO 1 gemappt.

Die Tabelle beschreibt das Mapping des PDO statusword

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
01_h	60410010_h	statusword

IV.3.2.1.2 PDO ASCII-Kanal senden (21)

Sobald ASCII-Zeichen in den ASCII Sendepuffer übertragen werden, werden diese mit Hilfe dieses PDOs (Default-PDO) zum Master (Steuerung) übertragen. Das erfolgt immer dann, wenn mit Hilfe des PDO ASCII-Kanal empfangen (⇒ Kap. IV.3.1.1.2) Kommandos bzw. Parameter übertragen worden sind. Nach dem Einschalten ist dieses PDO auf TPDO2 gemappt.

Die Tabelle beschreibt das Mapping des PDO ASCII-Kanal senden

Subindex	Wert	Beschreibung
00_h	8_h	Anzahl der Einträge
01_h..08_h	31020208_h	1. bis 8. ASCII-Zeichen

Dieses Objekt unterstützt **nur** den Übertragungstyp 255 (asynchron).

IV.3.2.1.3 PDO Ist-Lage (22)

Das PDO Ist-Lage besteht aus statusword (UNSIGNED16), Istlage (UNSIGNED24) und Umdrehung pro Minute (UNSIGNED24). Mit diesem PDO kann die Position im Mode “Drehzahl digital“ oder “Strom digital“ ermittelt werden.

Die Tabelle beschreibt das Mapping des PDO Ist-Lage:

Subindex	Wert	Beschreibung
00_h	3_h	Anzahl der Einträge
01_h	60410010_h	statusword
02_h	20700118_h	Istlage (20 Bit / Umdrehung)
03_h	20700218_h	Drehzahl*

* Auflösung: 1 Bit = 1875 / 262144 min^{-1}

IV.3.2.1.4 PDO Erweiterter Status (23)

Das PDO Erweiterter Status besteht aus dem statusword (UNSIGNED16) und dem herstellerspezifischen Statusregiste(UNSIGNED32). Dieses PDO kann zusätzlich auf ein Ereignis im Bereich des Statusregisters getriggert werden. Dieses PDO ist modeunabhängig.

Die Tabelle beschreibt das Mapping des PDO Erweiterter Status:

Subindex	Wert	Beschreibung
00_h	2_h	Anzahl der Einträge
01_h	60410010_h	statusword
02_h	10020020_h	Herstellerspezifisches Statusregister

IV.3.2.1.5 PDO Ist-Lage 2 (32)

Das PDO Ist-Lage 2 ist ein zeit- und datenoptimiertes PDO (vgl. PDO 21). Es beinhaltet die Istlage (UNSIGNED24) und die Umdrehung pro Minute (UNSIGNED 24). Mit diesem PDO kann die Position im Mode “Drehzahl digital“ oder “Strom digital“ ermittelt werden.
Dieses PDO kann **ausschließlich** mit dem **SYNC - Objekt** angefordert werden.

Die Tabelle beschreibt das Mapping des PDO Ist-Lage 2:

Subindex	Wert	Beschreibung
00_h	2_h	Anzahl der Einträge
01_h	20700118_h	Istlage (20 Bit / Umdrehung)
02_h	20700218_h	Drehzahl *

* Auflösung: 1 Bit = 1875 / 262144 min^{-1}

Dieses Objekt unterstützt **nur** die Übertragungstypen 1 bis 240 (zyklisch synchron).

IV.3.2.1.6 PDO Inkrementeller Positionsistwert (33)

Das PDO Inkrementeller Positionsistwert ist ein datenoptimiertes Objekt, das **ausschließlich** mit einem **SYNC - Objekt** angefordert werden kann.

Berechnung der absoluten Position:

$$\text{Position} = \frac{\text{inkementeller Positionsistwert}}{2^{20}}$$

Die Tabelle beschreibt das Mapping des PDO Inkrementeller Positionsistwert:

Subindex	Wert	Beschreibung
00_h	2_h	Anzahl der Einträge
01_h	20700320_h	Inkrementeller Positionsistwert
02_h	20701120_h	erweitertes Statusregister für PDO 33

Dieses Objekt unterstützt **nur** die Übertragungstypen 1 bis 240 (zyklisch synchron).

IV.3.2.1.7 PDO Positionsschwellen (34)

Das PDO Positionsschwellen wird verwendet, um ein Überschreiten der schnellen Positionsregister P1..P16 zu erkennen. Sie werden über die SDOs 2051 - 2053 konfiguriert

Die Tabelle beschreibt das Mapping des PDO Positionsschwellen:

Subindex	Wert	Beschreibung
00_h	1_h	Anzahl der Einträge
01_h	20860010_h	Positionsschwellenanzeige

IV.3.2.1.8 PDO Frei definierbar (37 bis 40)

Sind dies PDOs angewählt können die Objekte frei hinzugefügt werden. Hierzu werden die SDOs $1A00_h$ bis $1A03_h$ (Mapping Parameter) verwendet. Es können bis 8 einzelne Objekte in ein PDO “gemappt” werden.



Achtung:
Auch wenn Objekte “event getriggert” sind und nicht über den CAN Bus gemeldet werden, werden diese dennoch überwacht. Das führt dazu, dass bei 16 gemappten Objekten die CPU Last sehr stark steigen kann. Um das zu verhindern, sollten wirklich nur die Objekte gemappt werden, die auch wirklich benötigt werden, oder die “inhibit Zeit” muss entsprechend hoch gesetzt werden (z.B. 400 ms). Die Folge einer zu hohen CPU – Last kann eine lange Reaktionszeit bei einem SDO Datendienst (Antwortzeit im Mittel > 40 ms) sein.

IV.3.2.2 Objektbeschreibung

IV.3.2.2.1 SDO 1800-1803h: 1st -4th Transmit - PDO Kommunikationsparameter (DS301)

Subindex	Beschreibung
00_h	Anzahl der Einträge
01_h	PDO COB-ID*
02_h	transmission type
03_h	inhibit time**
04_h	CMS priority group

* Nach der Selektion eines PDOs über $2A00_h$ bis $2A03_h$ kann hier der COB-Identifier benannt werden, auf den der Verstärker reagieren soll (falls ein abweichender Wert vom Defaultwert erforderlich ist)

** Hiermit wird eine Ruhezeit eingestellt, die nach dem Senden eines PDOs eingehalten werden muss, bevor es wieder gesendet werden kann (nur wichtig bei event-gesteuerten PDOs).

IV.3.2.2.2 SDO 1A00-1A03h: 1st -4th Transmit - PDO Mappingparameter (DS301)

Subindex	Beschreibung
00_h	Anzahl der konfigurierten Objekte
01_h..08_h	Beschreibung der konfigurierten Objekte

IV.3.2.2.3 SDO 2A00-2A03h: 1st -4th Transmit PDO select

Mit Hilfe dieses Objekts wird ein vordefiniertes Sende - PDO ausgewählt.
Mit den SDOs 1800_h-1803_h "1st -4th Transmit PDO parameter" und $1A00_h$-$1A03_h$ "1st -4th Transmit PDO mapping" können dann die Eigenschaften dieses PDOs festgelegt werden.

Dieses Objekt ermöglicht ein variables Mapping von vordefinierten PDOs. Die möglichen anwählbaren PDOs sind in der Tabelle in Kapitel IV.3.2 aufgeführt.

Objektbeschreibung:

Index	$2A00_h$ ...$2A03_h$
Name	1st -4th Transmit PDO select
Objektcode	VAR
Datentyp	UNSIGNED8

Wertbeschreibung:

Zugriff	r/w
PDO Mapping	nicht möglich
Wertebereich	0..255
Defaultwert	0

IV.3.2.2.4 SDO 2014-2017h: Maske 1-4 für Sende-PDOs

Um die Buslast bei ereignisgesteuerten PDOs zu senken, kann die Überwachung für einzelne Bits im PDO mit Hilfe der Masken abgeschaltet werden. Beispielsweise kann so auch erreicht werden, dass Positionsistwerte immer nur pro Umdrehung gemeldet werden.
Diese Objekte maskieren den PDO - Kanal 1 für TPDOs mit den Nummern 37 bis 40. Sollten nur 2 Byte in einem PDO definiert sein, überlagert die Maske auch nur zwei Byte obwohl 4 Byte Maskeninformation übertragen worden ist.
Ein aktiviertes Bit in der Maske bedeutet, dass die Überwachung für das korrespondierende Bit im PDO aktiv ist, d. h. überwacht wird.

Index	$0x2014_h$...$0x2017_h$.
Name	tx_mask 1-4
Objektcode	ARRAY
Number of elements	2
Datentyp	UNSIGNED32

Subindex	01_h
Kurzbeschreibung	tx_mask1-4_low
Mode	unabhängig
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	—
EEPROM	nein
Defaultwert	$FFFFFFFF_h$

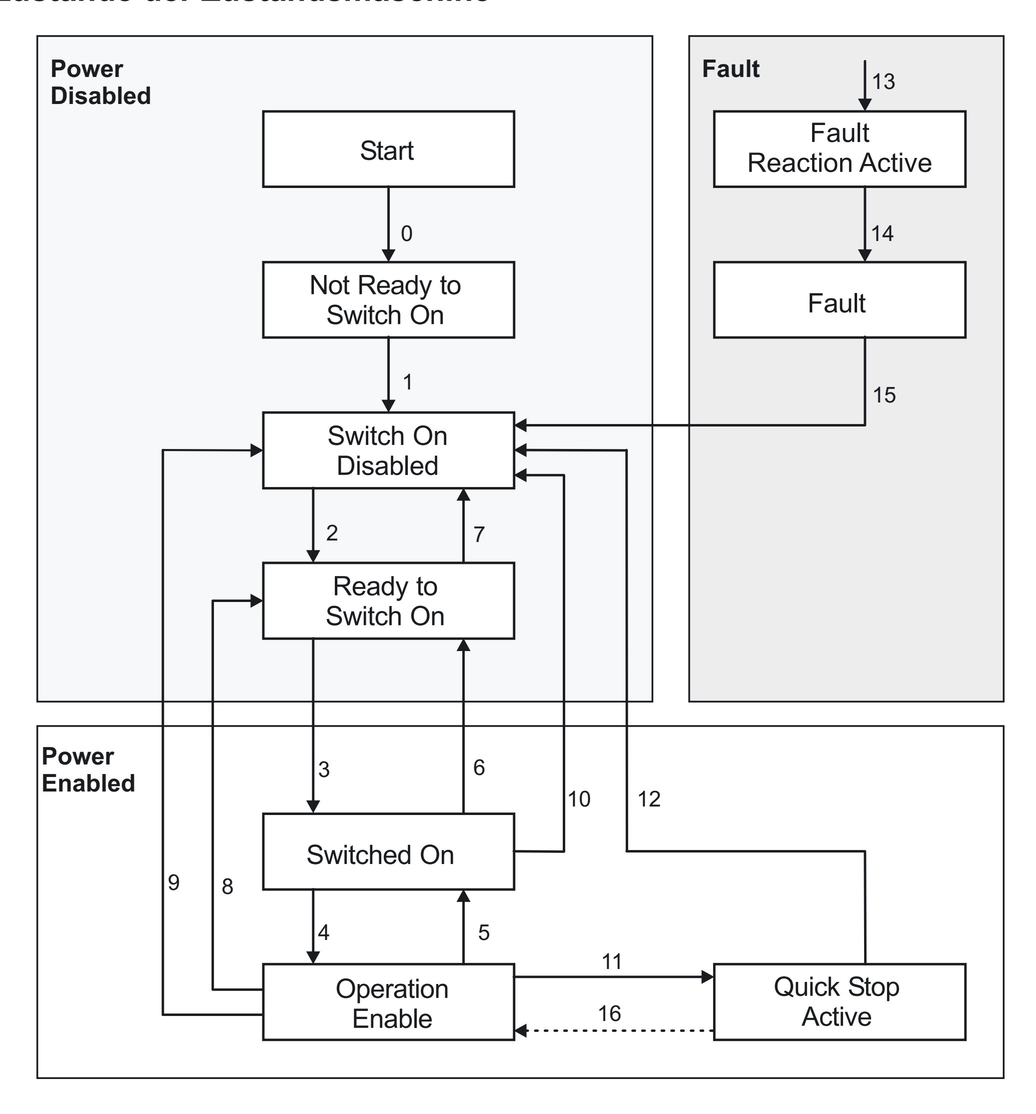
Subindex	02_h
Kurzbeschreibung	tx_mask1-4_high
Mode	unabhängig
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	—
EEPROM	nein
Defaultwert	$FFFFFFFF_h$

IV.4 Gerätesteuerung (dc)

Mit Hilfe der SERVOSTAR Gerätesteuerung können sämtliche Fahrfunktionen in den entsprechenden Modi ausgeführt werden. Die Steuerung des SERVOSTAR ist über eine modeabhängige Zustandsmaschine realisiert. Zur Steuerung der Zustandsmaschine dient das controlword (⇒ Kap. IV.4.2.1).
Die Modeeinstellung erfolgt über das Objekt “Modes of Operation“ (⇒ Kap. VI.3). Die Zustände der Zustandsmaschine können mit dem statusword ermittelt werden (⇒ Kap. IV.4.2.2).

IV.4.1 Zustandsmaschine (DS402)

IV.4.1.1 Zustände der Zustandsmaschine



Zustand	Beschreibung
Not Ready to Switch On	SERVOSTAR ist nicht einschaltbereit, es wird keine Betriebsbereitschaft (BTB) vom Reglerprogramm gemeldet.
Switch On Disable	SERVOSTAR ist einschaltbereit, Parameter können übertragen werden, Zwischenkreisspannung kann eingeschaltet werden, Fahrfunktionen können noch nicht ausgeführt werden
Ready to Switch On	Zwischenkreisspannung kann angeschaltet werden. Parameter können übertragen werden, Fahrfunktionen können noch nicht ausgeführt werden
Switched On	Zwischenkreisspannung muss angeschaltet sein, Parameter können übertragen werden, Fahrfunktionen können noch nicht ausgeführt werden, Endstufe ist eingeschaltet (Enabled)
Operation Enable	kein Fehler steht an, Endstufe ist eingeschaltet (Enabled), Fahrfunktionen sind “enabled“
Quick Stop Active	Antrieb ist mit der Notbremsrampe gestoppt worden, Endstufe ist eingeschaltet (Enabled), Fahrfunktionen sind “enabled”, Verhalten abhängig von SDO $605A_h$ (⇒ Kap. VI.3)
Fault Reaction Active	wird z.Zt. nicht unterstützt
Fault	wird z.Zt. nicht unterstützt

IV.4.1.2 Übergänge der Zustandsmaschine

Die Zustandsübergänge werden durch interne Ereignisse (z.B. Ausschalten der Zwischenkreisspannung) und durch die Flags im controlword (Bits 0,1,2,3,7) beeinflusst.

Übergang	Ereignis	Aktion
0	Reset	Initialisierung
1	Initialisierung erfolgreich abgeschlossen. SERVOSTAR betriebsbereit	keine
2	Bit 1 (disable Voltage) und Bit 2 (Quick Stop) im controlword gesetzt (‘Shutdown’ command). Zwischenkreisspannung kann anliegen.	keine
3	Bit 0 wird zusätzlich gesetzt (‘Switch On’ command)	Endstufe wird eingeschaltet (enable), sofern Hardware-enable anliegt (UND-Verknüpfung). Antrieb hat Drehmoment.
4	Bit 3 wird zusätzlich gesetzt (‘Enable Operation’ command)	Fahrfunktion in Abhängigkeit des eingestellten Mode wird freigegeben
5	Bit 3 wird gelöscht (‘Disable Operation’ command)	Fahrfunktion wird gesperrt. Antrieb wird mit der relevanten Rampe (modeabhängig) gestoppt. Aktuelle Position wird gehalten
6	Bit 0 wird gelöscht (‘Shutdown’ command)	Endstufe wird abgeschaltet (disable). Motor wird drehmomentfrei
7	Bit 1/2 werden gelöscht (’Quickstop’ / ‘Disable Voltage’ command)	keine
8	Bit 0 wird gelöscht (‘Shutdown’ command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
9	Bit 1 wird gelöscht (‘Disable Voltage’ command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
10	Bit 1/2 werden gelöscht (’Quickstop’ / ‘Disable Voltage’ command)	Fahrfunktion in Abhängigkeit des eingestellten Mode wird freigegeben.
11	Bit 2 wird gelöscht (’Quickstop’ command)	Antrieb wird mit der Notbremsrampe angehalten. Die Endstufe bleibt “enabled“. Sollwerte werden gelöscht (Fahrsatznummer, digitaler Sollwert, Geschwindigkeit für Tippbetrieb oder Referenzfahrt). Vor dem erneuten Ausführen einer Fahrfunktion muss Bit 2 wieder gesetzt werden
12	Bit 1 wird gelöscht (‘Disable Voltage’ command)	Endstufe wird abgeschaltet (disable) Motor wird drehmomentfrei
13	Wird zur Zeit nicht unterstützt	keine
14	Wird zur Zeit nicht unterstützt	keine
15	Wird zur Zeit nicht unterstützt	keine
16	Bit 2 wird gesetzt	Fahrfunktion ist wieder freigegeben.



Achtung
wenn der Servoverstärker über das controlword / statusword bedient wird, dürfen keine Steuerkommandos über einen anderen Kommunikationskanal (RS232, CANopen, ASCII-Kanal, Optionskarte) gesendet werden.

IV.4.2 Objektbeschreibung

IV.4.2.1 SDO 6040h: Controlword (DS402)

Aus der logischen Verknüpfung der Bits im controlword und externen Signalen (Freigabe der Endstufe) ergeben sich die Steuerkommandos. Die Definition der Bits wird im Folgenden beschrieben:

Index	$0x6040_h$
Name	Controlword
Objektcode	VAR
Datentyp	UNSIGNED16

Mode	all
Zugriff	r/w
PDO Mapping	möglich
Einheit	—
Wertebereich	0..65535
EEPROM	nein
Defaultwert	0

Bitbelegung im controlword

Bit	Name	Bit	Name
0	Switch on	8	Halt (Zwischenstopp)
1	Disable Voltage	9	reserved
2	Quick Stop	10	reserved
3	Enable Operation	11	Schleppfehler und Ansprechüberwachung quittieren
4	Operation mode specific	12	Rücksetzen der Position
5	Operation mode specific	13	Manufacturer specific
6	Operation mode specific	14	Manufacturer specific
7	Reset Fault (Nur wirksam bei Fehlern)	15	Manufacturer specific

Kommandos im controlword

Command	Bit 7 Fault Reset	Bit 3 Enable Operation	Bit 2 Quick Stop	Bit 1 Disable Voltage	Bit 0 Switch on	Übergänge
Shutdown	X	X	1	1	0	2, 6, 8
Switch on	X	X	1	1	1	3
Disable Voltage	X	X	X	0	X	7, 9, 10, 12
Quick Stop	X	X	0	1	X	7, 10, 11
Disable Operation	X	0	1	1	1	5
Enable Operation	X	1	1	1	1	4, 16
Fault Reset	1	X	X	X	X	15

Mit X gekennzeichnete Bits sind irrelevant.

Modeabhängige Bits im controlword

Die folgende Tabelle beschreibt die modeabhängigen Bits im controlword. Es werden z. Zt. nur herstellerspezifische Modi unterstützt. Die einzelnen Modi werden mit dem SDO 6060_h “Modes of operation” eingestellt.

Operation mode	Nr.	Bit 4	Bit 5	Bit 6
Lage	FF_h	reserviert	reserviert	reserviert
Drehzahl digital	FE_h	reserviert	reserviert	reserviert
Strom digital	FD_h	reserviert	reserviert	reserviert
Drehzahl analog	FC_h	reserviert	reserviert	reserviert
Strom analog	FB_h	reserviert	reserviert	reserviert
Trajektorie	FA_h	reserviert	reserviert	reserviert
Referenzierung	$F9_h$	Referenzfahrt starten	reserviert	reserviert
Tippbetrieb	$F8_h$	reserviert	reserviert	reserviert
Profile Position Mode (pp)	01_h	new_set_point	change_set_immediately	absolut / relativ
Profile Velocity Mode (pv)	03_h	reserviert	reserviert	reserviert
Homing Mode (hm)	06_h	homing_operation_start	reserviert	reserviert

Beschreibung der restlichen Bits im controlword

Im Folgenden werden die restlichen Bits des controlword beschrieben.

Bit 8 Zwischenstopp Ist Bit 8 gesetzt, wird in sämtlichen Modi der Antrieb gestoppt (Zwischenstopp). Die Sollwerte (Geschwindigkeit für Referenzfahrt oder Tippbetrieb, Fahrauftragsnummer, Sollwerte für Digitalmode) der einzelnen Modi bleiben erhalten
Bit 9,10 Diese Bits sind durch das Antriebsprofil (DS402) reserviert.
Bit 11 Fehler quittieren Durch Setzen von Bit 11 wird die Ansprechüberwachung und/oder der Schleppfehler quittiert.
Bit 12 Rücksetzen der Position unter Berücksichtigung des Referenzoffsets. (siehe auch Referenzfahrtart Nr. 6 in SDO 2024_h, Subindex 01_h)

Bit 13, 14, 15 Diese Bits sind herstellerspezifisch und z.Zt. reserviert.

IV.4.2.2 SDO 6041h: Statusword (DS402)

Der momentane Zustand der Zustandsmaschine kann mit Hilfe des statusword (⇒ Kap. IV.3.2.1.1) abgefragt werden.

Index	$0x6041_h$
Name	Statusword
Objektcode	VAR
Datentyp	UNSIGNED16

Mode	all
Zugriff	r/w
PDO Mapping	möglich
Einheit	—
Wertebereich	0..65535
EEPROM	ja
Defaultwert	0

Bitbelegung im statusword

Bit	Name	Bit	Name
0	Ready to switch on	8	Manufacturer specific (reserviert)
1	Switched on	9	Remote (nicht unterstützt)
2	Operation enable	10	Target reached
3	Fault (i.V.)	11	Internal limit active (nicht unterstützt)
4	Disable voltage	12	Operation mode specific (reserviert)
5	Quick stop	13	Operation mode specific (reserviert)
6	Switch on disabled	14	Manufacturer specific (reserviert)
7	Warning	15	Manufacturer specific (reserviert)

Zustände der Statusmaschine

State	Bit 6 switch on disable	Bit 5 quick stop	Bit 3 fault	Bit 2 operation enable	Bit 1 switched on	Bit 0 ready to switch on
Not ready to switch on	0	X	0	0	0	0
Switch on disabled	1	X	0	0	0	0
Ready to switch on	0	1	0	0	0	1
Switched on	0	1	0	0	1	1
Operation enabled	0	1	0	1	1	1
Fault	Wird z. Zt. nicht unterstützt					
Fault reaction active	Wird z. Zt. nicht unter-stützt	X	X	X	X	15
Quick stop active	0	0	0	1	1	1

Die mit X gekennzeichneten Bits sind irrelevant

Beschreibung der restlichen Bits im statusword

Bit 4: voltage_disable Die Zwischenkreisspannung liegt an, wenn das Bit gelöscht ist.

Bit 7: warning Für das Setzen von Bit 7 kann es mehrere Gründe geben, die zu dieser Warnung geführt haben. Der Grund für diese Warnung kann anhand des SDO 1002_h “herstellerspezifisches Statusregister“ ermittelt werden. (⇒ Kap. IV.2.1.3)

Bit 8: Toggle – Bit Mode Tippbetrieb, Referenzierung ,Lage, Positionierung (pp), Referenzierung (hm) Das Toggle - Bit wird immer dann verändert (setzen bzw. rücksetzen), wenn ein Fahrsatz erfolgreich ausgeführt wurde (Inkrement genau in Zielposition!). Das Bit wird nicht “getoggelt”, wenn ein Fahrsatz abgebrochen wird (z.B.: mit dem STOP – Kommando oder beim Schleppfehler). Die Auswertung des Toggle - Bits kann in Kombination mit dem Bit 10 “target reached” (SDO 6041_h) und “Fahrsatz aktiv” Bit 16 (SDO 1002_h) erfolgen. Die Auswertung des Bits ist dann sinnvoll, wenn aufgrund der Fahrsatzdaten keine Änderung des Bits 10 oder Bits 16 sichtbar werden (sehr kurze oder gleiche Fahrsätze).

Bit 9: remote Wird nicht unterstützt

Bit 10: target_reached Wird gesetzt, wenn der Antrieb die Zielposition erreicht hat.

Bit 11: internal_limit_active Wird nicht unterstützt

IV.4.2.3 SDO 6060h: modes_of_operation (DS402)

Mit diesem Objekt wird der Mode eingestellt und mit dem SDO 6061_h kann der eingestellte Mode gelesen werden. Es können zwei Arten von Operationsmodi unterschieden werden:

herstellerspezifische Operationsmodi
Diese Operationsmodi sind auf die Gerätefunktionalitäten optimierte Betriebszustände.

Operationsmodi nach CANopen Antriebsprofil DSP402
Diese Operationsmodi nach CANopen Antriebsprofil DSP402 definiert.

Nach einem Modewechsel muss der entsprechende Sollwert neu gesetzt werden (z.B.: Mode Referenzierung _ Sollwert Referenzfahrtgeschwindigkeit).Wurde der Mode Lage oder Tippbetrieb gespeichert, so ist der Mode Referenzierung nach einem RESET des Servoverstärkers eingestellt.

Ein Operationmode ist erst gültig, wenn er mit SDO 6061_h gelesen werden konnte.



ACHTUNG:
Niemals den Modus bei drehendem Motor umschalten!
Das Umschalten der Modi ist bei “enabletem” Verstärker grundsätzlich nur bei Drehzahl 0 erlaubt. Setzen Sie vor dem Umschalten den Sollwert auf 0.

Index	$0x6060_h$
Name	mode_of_operation
Objektcode	VAR
Datentyp	INTEGER8

Mode	all
Zugriff	w
PDO Mapping	möglich
Einheit	—
Wertebereich	—
EEPROM	nein
Defaultwert	—

Mode	mode dez.	mode hex.	Kommentar
	-10..-128	$F8_h..80_h$	reserviert
elektr. Getriebe	-9	$F7_h$	—
Tippbetrieb	-8	$F8_h$	—
Referenzierung	-7	$F9_h$	—
Trajektorie	-6	FA_h	—
Strom analog	-5	FB_h	—
Drehzahl analog	-4	FC_h	—
Strom digital	-3	FD_h	—
Drehzahl digital	-2	FE_h	—
Lage	-1	FF_h	Notwendiger Mode für Fahraufträge
—	0	0	reserviert nach DSP402
Positionierung (pp)	1	1_h	nach DSP402
Drehzahl (vl)	2	2_h	wird nicht unterstützt
Drehzahl (pv)	3	3_h	nach DSP 402
Drehmoment (tq)	4	4_h	nach DSP 402 (wird z. Zt. nicht unterstützt)
—	5	5_h	reserviert
Referenzierung (hm)	6	6_h	nach DSP 402
Interpolation	7	7_h	nach DSP 402 (wird nicht unterstützt)
—	8..127	$8_h..7F_h$	reserviert

IV.4.2.4 SDO 6061h: mode_of_operation_display (DS402)

Mit diesem Objekt kann der über das SDO 6060_h eingestellte Mode gelesen werden. Ein Operationmode ist erst gültig, wenn er mit SDO 6061_h gelesen werden konnte (s. auch SDO 6060_h).

Index	$0x6061_h$
Name	mode_of_operation_display
Objektcode	VAR
Datentyp	INTEGER8

Mode	all
Zugriff	w
PDO Mapping	möglich
Einheit	—
Wertebereich	—
EEPROM	nein
Defaultwert	—

IV.5 Factor Groups (fg) (DS402)

IV.5.1 Allgemeine Informationen

IV.5.1.1 Faktoren

Es gibt eine Umrechnungsmöglichkeit für die physikalischen Dimensionen und Größen auf die antriebsinternen Einheiten (Inkremente). Dafür können einige Faktoren implementiert werden. Dieses Kapitel beschreibt den Einfluss dieser Faktoren auf das System, die Berechnungsvorschriften für diese Faktoren und die Daten, die dazu erforderlich sind.
Normierte Parameter sind mit * gekennzeichnet.

IV.5.1.2 Beziehung zwischen physikalischen und antriebsinternen Einheiten

Die in den factor groups festgelegten Faktoren definieren die Beziehung zwischen antriebsinternen Einheiten (Inkrementen) und physikalischen Einheiten.

Die Faktoren sind das Ergebnis einer Berechnung von zwei Parametern: des dimension index und des notation index. Der dimension index beschreibt dabei die physikalische Dimension, der notation index die physikalische Einheit und den Dezimalpunkt für Werte. Diese Faktoren werden zur Normierung der physikalischen Werte verwendet.

Der notation index kann auf zwei Arten verwendet werden:

- Für eine Einheit mit dezimaler Skalierung und notation index < 64 definiert der notation index den Dezimalpunkt der Einheit.
- Für eine Einheit mit nicht-dezimaler Skalierung und notation index > 64 definiert der notation index den Subindex der physikalischen Dimension der Einheit.

IV.5.2 Objektbeschreibung

IV.5.2.1 SDO 608Bh: velocity_notation_index (DS402)

Index	$0x608B_h$
Name	velocity_notation_index
Objektcode	VAR
Datentyp	INTEGER8

Mode	all
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	-128..127
EEPROM	nein
Defaultwert	0

Dieses Objekt legt die Notation für das SDO 6081_h “profile_velocity” fest. In Kombination mit dem SDO $608C_h$ “velocity_dimension_index” können die folgenden Basiseinheiten dargestellt werden:

physical dimension	units	velocity_dimension_index	velocity_notation_index
herstellerspezifisch	inc/s	0	0
Umdrehungen	U/min	11	73

IV.5.2.2 SDO 608Ch: velocity_dimension_index (DS402)

Index	$0x608C_h$
Name	velocity_dimension_index
Objektcode	VAR
Datentyp	UNSIGNED8

Mode	all
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	0..255
EEPROM	nein
Defaultwert	0

Dieses Objekt legt die Dimension für das SDO 6081_h “profile_velocity” fest. In Kombination mit dem SDO $608B_h$ “velocity_notation_index” können die folgenden Basiseinheiten dargestellt werden:

physical dimension	units	velocity_dimension_index	velocity_notation_index
herstellerspezifisch	inc/s	0	0
Umdrehungen	U/min	11	73

IV.5.2.3 SDO 6093h: position_factor (DS402)

Der Positionsfaktor (position_factor) konvertiert die gewünschte Position in das interne Datenformat des SERVOSTAR (Inkremente).
Die Auflösung des Lagereglers kann mit 20 Bit/Umd oder 16 Bit/Umd betrieben werden (s. SDO $35D1_h$ und ASCII – Kommando PRBASE). Mit Hilfe des Numerators und der Vorschubkonstante kann jede beliebige Skalierung eingestellt werden.

$$position_factor = \frac{position_encoder_resolution \times gear_ratio}{feed_constant}$$

- **position_encoder_resolution**
  Auflösung des Lagereglers 2^{20} oder 2^{16}
- **gear_ratio**
  Übersetzungsverhältnis des eingesetzten Getriebes
- **feed_constant**
  Vorschubkonstante des Abtriebes

Index	$0x6093_h$
Name	position_factor
Objektcode	ARRAY
Number of elements	2
Datentyp	UNSIGNED32

Subindex	01_h
Kurzbeschreibung	numerator
Mode	pp
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	Inkremente [inc]
Wertebereich	$0..(2^{32}-1)$
EEPROM	nein
Defaultwert	2^{20}

Subindex	02_h
Kurzbeschreibung	feed_constant
Mode	pp
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	$0..(2^{32}-1)$
EEPROM	nein
Defaultwert	10000

Beispiel: Eine Umdrehung soll 10000 Inkrementen entsprechen. Übersetzungsverhältnis ist 1.

$$position_factor = \frac{2^{20} Inkr.}{10000 Inkr.}$$

⇒ **Numerator**: 2^{20} **Vorschubkonstante**: 10000
⇒ Sollwertvorgabe in [Inkr / Umd.] für das SDO $607A_h$ (target_position)

Der Numerator entspricht im Gerät dem ASCII - Parameter PGEARO, die feed_constant dem Parameter PGEARI.

IV.5.2.4 SDO 6094h: velocity_encoder_factor (DS402)

Der velocity_encoder_factor konvertiert die gewünschte Drehzahl (Umd/min) oder Geschwindigkeit (inc/s) in das interne Datenformat des SERVOSTAR (Inkremente).

Für die Berechnung des velocity_encoder_factor gilt:

$$\text{velocity_encoder_factor} = \frac{\text{velocity_encoder_resolution} \times \text{gear_ratio} \times \text{position_unit} \times \text{Fvelocity(notation_index)}}{\text{feed_constant} \times \text{velocity_unit} \times \text{sec} \times \text{Fposition(notation_index)}}$$

- **velocity_encoder_resolution** Auflösung des Drehzahlreglers 2^{20}
- **gear ratio** Übersetzungsverhältnis des eingesetzten Getriebes
- **position unit** in Meter
- **Fposition(notation_index)** in dimension_index = 1, notation_index = 0 [m]
- **feed_constant** Vorschubkonstante des Abtriebes
- **velocity_unit** in [m/s]
- **Fvelocity (notation_index)** in dimension_index = 13, notation_index = 0 [m / s]

(s. auch SDO 606B$_h$ “velocity_notation_index” und SDO 606C$_h$ “velocity_dimension_index”)

Index	0x6094$_h$
Name	velocity_encoder_factor
Objektcode	ARRAY
Number of elements	2
Datentyp	UNSIGNED32

Subindex	01$_h$
Kurzbeschreibung	numerator
Mode	pv
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	Inkremente [inc]
Wertebereich	0..(2^{32}-1)
EEPROM	nein
Defaultwert	0

Subindex	02$_h$
Kurzbeschreibung	divisor
Mode	pv
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	Sekunde [s]
Wertebereich	0..(2^{32}-1)
EEPROM	nein
Defaultwert	0

Beispiel: Die Drehzahl - Sollwertvorgabe soll in Umdrehungen pro Minute erfolgen. Das Übersetzungsverhältnis und die Vorschubkonstante ist 1.

$$velocity_encoder_factor = \frac{2^{20}}{1} \quad \frac{1/1}{1} \quad \frac{\text{Inkr.}}{\text{s}}$$

Erweiterung auf Umdrehung / min:

$$velocity_encoder_factor = \frac{2^{20}}{1} \quad \frac{1/1}{1} \quad \frac{[\text{m}]}{[\text{m/s}]} \quad \frac{\text{Inkr.}}{\text{s}} \times \frac{1}{60[\text{s/min}]} = \frac{2^{20}}{60} \quad \frac{\text{Inkr.}}{\text{s}}$$

⇒ **Numerator**: 2^{20} **Divisor**: 60

⇒ Sollwertvorgabe in [Umd / min] für das SDO $60FF_h$ "target_velocity / velocity units"

⇒ Sollwertvorgabe in [inc. / s] für das SDO 6081_h "Profile_velocity / speed units"

Anmerkung: Da der Drehzahlregler intern, unabhängig von der Auflösung des Gebersystems mit einer Auflösung von 2^{20} Bit/Umdrehung arbeitet, ergibt sich folgende Berechnung für den Operationsmode pv (für Umdrehung pro Minute):

$$\text{Inkremente} = \frac{262144}{1875} \times \text{Drehzahlsollwert}[\text{min}^{-1}]$$

Diese inkrementelle Sollwertvorgabe sollte bei zyklischen Anwendungen (z.B. Lageregelung, 4ms Zyklus) verwendet werden (Vorteile: keine Rundungsfehler, geringere CPU - Last). Die o. g. Umrechnung ist gültig, wenn **Divisor oder Numerator zu 0** gesetzt werden.
Der velocity_encoder_factor beeinflusst auch das SDO 6081_h “profile_velocity”. Um diesen Faktor auch für den Positioniermode (pp) verwenden zu können, müssen die internen Getriebefaktoren PGEARI und PGARO gleich sein (PGEARI =PGEARO; SDO 2020_h Subindex: 08_h,09_h). Werden **Divisor oder Numerator zu 0** gesetzt, wird die interne Skalierung “Inkremente pro Abtastzykuls (250 µs)” verwendet.

IV.5.2.5 SDO 6097h: acceleration_factor (DS402)

Der Beschleunigungsfaktor (acceleration_factor) konvertiert die Beschleunigung [Einheit /s^2] in das interne Format des SERVOSTAR.
Zur Zeit sind Numerator und Divisor nur lesbar. Die Werte sind zu 1 gesetzt. Ist der Beschleunigungsfaktor 1, werden die Rampenvorgaben (SDO 6083_h "profile_acceleration" und SDO 6084_h "profile_deceleration") als Beschleunigungszeit [ms] bezogen auf die Zielgeschwindigkeit (SDO 6081_h "profile_velocity") vorgegeben.

Index	$0x6097_h$
Name	acceleration_factor
Objektcode	ARRAY
Number of elements	2
Datentyp	UNSIGNED32

Subindex	01_h
Kurzbeschreibung	numerator
Mode	pp, pv
Zugriff	r
PDO Mapping	nicht möglich
Einheit	Inkremente [inc]
Wertebereich	0..(2^{32}-1)
EEPROM	nein
Defaultwert	0

Subindex	02_h
Kurzbeschreibung	divisor
Mode	pp, pv
Zugriff	r
PDO Mapping	nicht möglich
Einheit	—
Wertebereich	0..(2^{32}-1)
EEPROM	nein
Defaultwert	0

IV.6 Herstellerspezifischer Strom- und Drehzahlmode

IV.6.1 SDO 2060h: Digitaler Strom- oder Drehzahlsollwert

Index	2060_h
Kurzbeschreibung	Digitaler Strom- oder Drehzahlsollwert
Einheit	A bzw. min^{-1}
Zugriff	r/w
PDO mapping	möglich (vorgemappt auf wählbares RPDO 22)
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung:

Dieses Objekt dient zur Übergabe digitaler Sollwerte, die abhängig vom eingestellten Digitalmode (Mode FD_h = Strom digital, Mode FE_h = Drehzahl digital, einstellbar über SDO 6060_h) ausgewertet werden. Die Normierungen sind dabei wie folgt festgelegt:

Strom: $I[A] = \text{digitalerStromsollwert} / 1640 \times I_{max}$

Drehzahl: $n[min^{-1}] = 1875 / 262144 \times \text{digitaler Drehzahlsollwert}$

Ein neuer Sollwert wird immer erst nach einem erneuten Enable Operation gültig.
(über SDO 6040_h, Controlword)



Der SERVOSTAR - Lageregler ist bei aktiver Drehzahl- oder Stromregelung abgeschaltet.

IV.7 Einrichtdaten für den herstellerspez. Tipp- und Referenziermode

IV.7.1 SDO 2024h: Einrichtbetrieb für den Mode Large (SERVOSTAR)

Index	2024_h
Kurzbeschreibung	Parameter für Referenzfahrt und Tippen
Objektcode	RECORD
Number of elements	7

Beschreibung:

Über diesen Index werden Parameter eingegeben, die für die Betriebsarten Referenzieren und Tippen wichtig sind.

Beschreibung der Subindizes:

Subindex	01_h
Kurzbeschreibung	Referenzfahrtart
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0..6
Defaultwert	0

Beschreibung: Über diesen Index wird die Art der Referenzfahrt eingestellt. Dabei sind die folgenden Einstellungen möglich:

Wert	Bedeutung
0	Der Referenzpunkt wird an der Ist - Position gesetzt. Die dann gemeldete Istposition ist gleich dem eingestellten Referenzoffset.
1	Referenzfahrt auf Referenzschalter mit anschliessender Suche des Resolvernullpunkts.
2	Referenzfahrt auf Endschalter mit anschliessender Suche des Resolvernullpunkts.
3	Referenzfahrt auf Referenzschalter ohne anschliessende Suche des Resolvernullpunkts.
4	Referenzfahrt auf Endschalter ohne anschliessende Suche des Resolvernullpunkts.
5	Referenzfahrt innerhalb einer Motorumdrehung bis zum Resolvernullpunkt. Die Fahrtrichtung wird dabei durch den Subindex 02_h vorgegeben. Dabei bedeuten: 0: Fahrtrichtung negativ 1: Fahrtrichtung positiv 2: Motordrehung in Richtung der kürzesten Entfernung zum Resolvernullpunkt innerhalb einer Umdrehung.
6	Der Referenzpunkt wird an der aktuellen Lageregler-Sollposition auf den Wert des Referenzoffsets gesetzt. Die neue Istposition behält den selben Abstand zur Sollposition wie zuvor.

Dabei ist zu beachten:
Für die Referenzfahrten 1 und 3 muss ein digitaler Eingang als Nullposition - Eingang (Home - Position) konfiguriert sein. Für die Referenzfahrten 2 und 4 muss ein digitaler Eingang als Hardware - Endschalter konfiguriert werden. Bei den Referenzfahrtarten 1 – 5 wird die Einstellung des Nullimpulsoffsets für die ROD - Ausgabe berücksichtigt (ASCII - Kommando ENCZERO) d.h. der Nullpunkt wird so gelegt, dass sowohl die Ausgabe des Nullimpulses als auch die Anzeige der 0-Position an der Stelle des Nullimpulsoffsets erfolgen.

Subindex	02_h
Kurzbeschreibung	Referenzfahrtrichtung
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0..2
Defaultwert	0

Beschreibung: Über diesen Index wird die Fahrtrichtung für die Referenzfahrtarten 1 bis 5 vorgegeben. Dabei haben die Werte die folgende Bedeutung:
0: Fahrtrichtung negativ
1: Fahrtrichtung positiv
2: Motordrehung in Richtung der kürzesten Entfernung zum Resolvernullpunkt innerhalb einer Umdrehung. (Nur relevant bei Referenzfahrtart 5).

Subindex	03_h
Kurzbeschreibung	Geschwindigkeit der Referenzfahrt
Einheit	µm/s
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index wird die Referenzfahrtgeschwindigkeit vorgegeben.

Subindex	04_h
Kurzbeschreibung	Beschleunigungsrampe Tippen / Referenzfahrt
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	1..32767
Defaultwert	10

Beschreibung: Über diesen Index wird die Beschleunigungsrampe für Referenzfahrt und Tippbetrieb eingestellt. Sie wird als Trapezrampe ausgeführt. Die eingestellte Zeit bezieht sich auf die eingestellten Geschwindigkeiten für Referenzfahrt und Tippen.

Subindex	05_h
Kurzbeschreibung	Bremsrampe für Tippen / Referenzfahrt
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	1..32767
Defaultwert	10

Beschreibung: Über diesen Index wird die Bremsrampe für Referenzfahrt und Tippbetrieb eingestellt. Sie wird als Trapezrampe ausgeführt. Die eingestellte Zeit bezieht sich auf die eingestellten Geschwindigkeiten für Referenzfahrt und Tippen. Bei den Referenzfahrten auf Hardwareendschalter wird als Bremsrampe die Notrampe (ASCII – Parameter DECSTOP) verwendet.

Subindex	06_h
Kurzbeschreibung	Referenzoffset
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index wird der Referenzoffset, d.h. der angezeigte Positionsistwert nach erfolgter Referenzierung (SDO 2070_h, Subindex 06_h) eingestellt.

Subindex	07_h
Kurzbeschreibung	Geschwindigkeit Tippbetrieb
Einheit	µm/s
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index wird die Tippgeschwindigkeit vorgegeben.

IV.8 Positionierdaten für den Mode Lage (SERVOSTAR)

IV.8.1 SDO 2020h: Lageregler

Index	2020_h
Kurzbeschreibung	Parameter für den Lageregler
Objektcode	RECORD
Number of elements	10

Beschreibung:

Über diesen Index werden alle allgemeinen Parameter für den Mode Lage vorgegeben.

Beschreibung der Subindizes:

Subindex	01_h
Kurzbeschreibung	Achstyp
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0, 1
Defaultwert	0

Beschreibung: Beschreibt die Art der mechanischen Achse.

Wert 0: Linearachse. Es gibt einen definierten Referenzpunkt, ab dem Positionen gezählt werden. Dieser muss durch eine Referenzfahrt oder durch Setzen eines Referenzpunktes definiert werden. Falls konfiguriert, wird die Bewegung der Achse durch Software-Endschalter begrenzt.

Wert 1: Rundachse. Es muss nicht referenziert werden. Beim Starten von Fahrsätzen oder des Tippbetriebes wird die Position jeweils auf 0 gesetzt Software-Endschalter begrenzen die Bewegung nicht.

Subindex	02_h
Kurzbeschreibung	In-Positionsfenster
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	4000_h

Beschreibung: Festlegung eines Zielfensters für Positionierungen. Wird die Zielfenstergrenze erreicht, wird im herstellerspezifischen Statusregister das Bit 19 gesetzt und bei entsprechender Ausgangskonfiguration der eingestellte Ausgang auf High gelegt.

Subindex	03_h
Kurzbeschreibung	Maximaler Schleppfehler
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	40000_h

Beschreibung: Festlegung eines Maximalwerts des Schleppfehlers. Überschreitet der auftretende Schleppfehler diesen Wert, wird der Antrieb gestoppt. Über das herstellerspezifische Statusregister wird im Bit 2 die Schleppfehlerüberschreitung angezeigt. Bei einem eingestellten Wert von 0 wird der Schleppfehler nicht überwacht.

Subindex	04_h
Kurzbeschreibung	Positionsregister 1
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 22 herstellerspezifisches Statusregister) gesetzt oder die Achse gestoppt. (Softwareendschalter 1 unterschritten = herstellerspezifisches Statusregister Bit 5 = 1).

Subindex	05_h
Kurzbeschreibung	Positionsregister 2
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 23 herstellerspezifisches Statusregister) gesetzt oder die Achse gestoppt. (Softwareendschalter 2 überschritten = herstellerspezifisches Statusregister Bit 6 = 1).

Subindex	06_h
Kurzbeschreibung	Positionsregister 3
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 24 herstellerspezifisches Statusregister) gesetzt.

Subindex	07_h
Kurzbeschreibung	Positionsregister 4
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Je nach Konfiguration wird bei Über- bzw. Unterschreitung der eingestellten Position ein Schwellwertbit (Bit 25 herstellerspezifisches Statusregister) gesetzt.

Subindex	08_h
Kurzbeschreibung	Auflösung / Nenner des Übersetzungsfaktors
Einheit	Umdrehungen
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	$1..(2^{32}-1)$
Defaultwert	1

Beschreibung: Über das Verhältnis der Werte der Subindizes 08_h und 09_h wird die mechanische Auflösung der Achse in µm / Umdrehung vorgegeben.

Subindex	09_h
Kurzbeschreibung	Auflösung / Zähler des Übersetzungsfaktors
Einheit	µm
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	$1..(2^{32}-1)$
Defaultwert	1

Beschreibung: Über das Verhältnis der Werte der Subindizes 08_h und 09_h wird die mechanische Auflösung der Achse in µm / Umdrehung vorgegeben.

Subindex	$0A_h$
Kurzbeschreibung	Zählrichtung
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	0, 1
Defaultwert	1

Beschreibung: Der Wert gibt die Zählrichtung für Strom-, Drehzahl- und Lageregelung vor. Bei einem Wert von 1 ist die positive Zählrichtung angewählt. Positive Sollwertvorgaben bewirken eine Drehung der Motorwelle im Uhrzeigersinn (bei Draufsicht).

IV.8.2 SDO 2022h: Positionierdaten für den Mode Lage

Index	2022_h
Kurzbeschreibung	Fahrauftragsparameter
Objektcode	RECORD
Number of elements	12

Beschreibung:

Über diesen Index werden alle Parameter eingegeben, die einen Bezug zu im Regler gespeicherten oder direkt ausführbaren Fahraufträgen haben. (Siehe ASCII-Kommando “ORDER”)

Beschreibung der Subindizes:

Subindex	01_h
Kurzbeschreibung	Position
Mode	Lage
Zugriff	r/w
PDO mapping	möglich (vorgemappt auf wählbares RPDO 34)
Datentyp	INTEGER32
Einheit	Inkremente oder µm
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
EEPROM	nein
Defaultwert	0

Beschreibung: Über diesen Index wird die Zielposition (Absolutfahrauftrag) oder die zu fahrende Strecke (Relativfahrauftrag) von Fahraufträgen vorgegeben. Dies wird durch Bit 0 der Fahrauftragsart festgelegt. Bit 13 der Fahrauftragsart legt fest, ob der angegebene Wert als Inkrement oder SI-Wert interpretiert werden soll.

Subindex	02_h
Kurzbeschreibung	Sollgeschwindigkeit gewichtet
Mode	Lage
Zugriff	r/w
PDO mapping	möglich (vorgemappt auf wählbares RPDO 34)
Datentyp	INTEGER16
Einheit	Inkremente/s oder µm/s
Wertebereich	-32768..32767
EEPROM	nein
Defaultwert	0

Beschreibung: Über diesen Index wird die Sollgeschwindigkeit für Fahraufträge vorgegeben. Er wird über Subindex $0D_h$ gewichtet. Wird der Wert durch Wahl von Fahrauftragsartbit 13 = 1 als SI-Einheit definiert, ergibt sich die Inkrementalgeschwindigkeit vi als $v_i = v_{SI} \times \frac{PGEARO}{PGEARI \times 4000}$,

wobei PGEARO (SDO 2020_h, Subindex 08_h) die Anzahl der Inkremente enthält, die gefahren werden, wenn die zu fahrende Strecke PGEARI (SDO 2020_h,Subindex 09_h) beträgt. Zu beachten ist, dass dabei eine Motorumdrehung einer Inkrementzahl von 2^{20} = 1048567 entspricht.

Subindex	03_h
Kurzbeschreibung	Fahrauftragsart
Mode	pp
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED8
Einheit	—
Wertebereich	0..$FFFF_h$
EEPROM	nein
Defaultwert	0

Beschreibung: Über diesen Index werden Fahrparameter des Fahrauftrags eingestellt. Die Bedeutung der Bits wird in den folgenden Tabellen erklärt.

Bit	Wert	Bedeutung
0	$0x0001_h$	Bit für die Art des relativen/absoluten Fahrsatzes (s. Tabelle 2)
1	$0x0002_h$	Bit für die Art des relativen Fahrsatzes (s. Tabelle 2)
2	$0x0004_h$	Bit für die Art des relativen Fahrsatzes (s. Tabelle 2)
3	$0x0008_h$	=0 kein Folgefahrsatz vorhanden, nach dem Erreichen der Zielposition bleibt der Antrieb stehen =1 Folgefahrsatz vorhanden, nach dem Erreichen der Zielposition wird automatisch der Folgefahrsatz gestartet. Die Nummer des Folgefahrsatzes wird mit dem Kommando O_FN vorgegeben.
4	$0x0010_h$	Bit für die Art des Folgefahrsatzes (s. Tabelle 3)
5	$0x0020_h$	Bit für die Art des Folgefahrsatzes (s. Tabelle 3)
6	$0x0040_h$	Bit für die Art des Folgefahrsatzes (s. Tabelle 3)
7	$0x0080_h$	Bit für die Art des Folgefahrsatzes (s. Tabelle 3)
8	$0x0100_h$	Bit für die Art des Folgefahrsatzes (s. Tabelle 3)
9	$0x0200_h$	reserviert
10	$0x0400_h$	reserviert
11	$0x0800_h$	reserviert
12	$0x1000_h$	=0 die Brems- und Anfahrbeschleunigung wird als Brems-/Anfahrzeit von 0 auf die Zielgeschwindigkeit (in msek) vorgegeben. =1 die Brems- und Anfahrbeschleunigung wird in mm/sek² vorgegeben. (s. auch Kommandos O_ACC1, O_ACC2, O_DEC1, O_DEC2).
13	$0x2000_h$	=0 die Zielposition und die Zielgeschwindigkeit des Fahrsatzes werden als Inkremente interpretiert. Es findet keine Umrechnung statt. =1 die Zielposition und die Zielgeschwindigkeit werden vor dem Start des Fahrsatzes in Inkremente umgerechnet. Für die Umrechnung werden die Parameter PGEARI und PGEARO benutzt. (s. auch Kommandos O_S, O_V, PGEARI, PGEARO)
14	$0x4000_h$	=0 die Geschwindigkeit des Fahrsatzes wird beim Fahrsatzstart als die Zielgeschwindigkeit übernommen. =1 die Zielgeschwindigkeit wird beim Fahrsatzstart analog vorgegeben (SW1). Beim Start des Fahrsatzes wird der analoge SW1-Wert eingelesen und als die Zielgeschwindigkeit übernommen (Skalierung: 10V=VSCALE1). Das Vorzeichen der SW1-Spannung wird ignoriert.
15	$0x8000_h$	Bit 3 für die Art des relativen Fahrsatzes (s. separate Tabelle)

Art des relativen / absoluten Fahrsatzes

Bit 0	Bit 1	Bit 2	Bit 15	Bedeutung
0	x	x	x	Absoluter Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als die Zielposition gewertet.
1	0	0	x	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. Die Zielposition wird abhängig vom Zustand der IN-POSITION Meldung berechnet: IN-POSITION=1: neue Zielposition=letzte Zielposition + Verfahrstrecke IN-POSITION=0: neue Zielposition=aktuelle Position + Verfahrstrecke
1	1	0	x	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=letzte Zielposition+Verfahrstrecke
1	0	1	x	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=aktuelle Position+Verfahrstrecke
1	1	1	0	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=positive Latchposition+Verfahrstrecke (s. auch Kommando LATCH32)
1	1	1	1	Relativer Fahrsatz, die Positionsangabe innerhalb des Fahrsatzes wird als Verfahrstrecke gewertet. neue Zielposition=negative Latchposition+Verfahrstrecke (s. auch Kommando LATCH32N)

Art des Folgefahrsatzes

Bit 4 NOBRAKE	Bit 5 FOL_IO	Bit 6 HI/LO	Bit 7 FTIME	Bit 8 VTARG	Bedeutung
0	0	0	0	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremst in die Zielposition des ersten Fahrsatzes. Anschließend wird der Folgefahrsatz gestartet.
1	0	0	0	0	Fliegendes Umschalten auf den Folgeauftrag in der Zielposition. Der Antrieb fährt mit der Zielgeschwindigkeit bis in die Zielposition des ersten Fahrsatzes. Anschließend wird bei voller Geschwindigkeit auf den Folgefahrsatz umgeschaltet.
1	0	0	0	1	Fliegendes Umschalten auf den Folgeauftrag in der Zielposition. Der Umschaltpunkt auf den Folgefahrauftrag wird so vorverlegt, dass in der Zielposition des ersten Fahrsatzes die Zielgeschwindigkeit des Folgefahrsatzes bereits erreicht wird.
0	1	0	0	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktion INxMODE=15) den Zustand LOW erreicht hat.
0	1	1	0	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktion INxMODE=15) den Zustand HIGH erreicht hat.
0	0	0	1	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremst in die Zielposition des ersten Fahrstazes. Der Folgefahrsatz wird gestartet, nachdem die programmierte Verzögerungszeit (O_FT) abgelaufen ist.
0	1	0	1	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktiopn INxMODE=15) den Zustand LOW erreicht hat oder nachdem die programmierte Verzögerungszeit (O_FT) abgelaufen ist.
0	1	1	1	0	Umschalten auf Folgefahrauftrag mit Abbremsen. Der Antrieb bremst in die Zielposition des ersten Fahrsatzes. Der Folgefahrsatz wird gestartet, wenn der dazu definierte Eingang (Funktiopn INxMODE=15) den Zustand HIGH erreicht hat oder nachdem die programmierte Verzögerungszeit (O_FT) abgelaufen ist.

Subindex	04_h
Kurzbeschreibung	Trajektorie
Einheit	—
Zugriff	r/w
PDO mapping	möglich (vorgemappt auf wählbares RPDO 33)
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: In Vorbereitung

Subindex	05_h
Kurzbeschreibung	Fahrauftragsnummer
Einheit	—
Zugriff	r/w
PDO mapping	möglich (vorgemappt auf wählbares RPDO 35)
Datentyp	UNSIGNED16
Wertebereich	1..180, 129..255
Defaultwert	0

Beschreibung: Über diesen Index wird die Nummer des angewählten Fahrauftrags vorgegeben. Dabei sind die Fahraufträge 1 bis 180 EEPROM – Fahrsätze, die Fahraufträge 192 bis 255 sind RAM – Fahraufträge. Die RAM – Fahraufträge werden beim Einschalten oder bei einem Reset des Servoverstärkers mit den ersten 64 EEPROM – Fahraufträgen gefüllt. Der Fahrauftrag 0 ist ebenfalls ein RAM – Fahrauftrag, der zum einen als Kopierpuffer für Fahraufträge zum anderen zum Eintragen von Fahrauftragsdaten für den Direktfahrauftrag (RPDO 34) Verwendung findet.

Subindex	06_h
Kurzbeschreibung	Anfahrzeit (Beschleunigung)
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	1..65535
Defaultwert	0

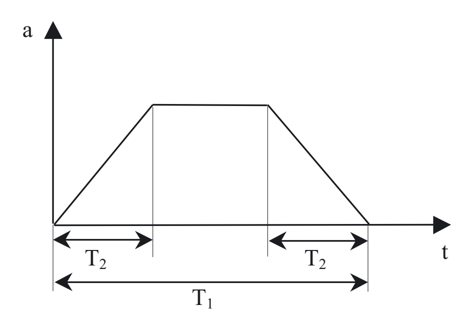
Beschreibung: Über diesen Index wird die Gesamtzeit angegeben, um auf die Zielgeschwindigkeit des Fahrauftrags zu kommen. Über die Wahl des Wertes des Subindex 08_h wird die Form der Beschleunigungsrampe eingestellt.

Subindex	07_h
Kurzbeschreibung	Bremszeit (Verzögerung)
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	1..65535
Defaultwert	0

Beschreibung: Über diesen Index wird die Gesamtzeit angegeben, um die Geschwindigkeit zur Zielposition auf 0 zu bringen. Über die Wahl des Wertes des Subindex 09_h wird die Form der Beschleunigungsrampe eingestellt.

Subindex	08_h
Kurzbeschreibung	Ruckbegrenzung (Beschleunigung) (i.V)
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	1..65535
Defaultwert	0

Beschreibung: Über diesen Index wird die Form der Beschleunigungsrampe vorgegeben. Der Wert muss kleiner als die Hälfte der Anfahrzeit (Subindex 06_h) gewählt werden. Folgende Darstellung dient zur Veranschaulichung des Sachverhalts:



Dabei entspricht T_1 dem Subindex 06_h, T_2 dem Subindex 08_h.Bei $T_2 = 0$ wird eine Trapezrampe gefahren, bei $T_2 = \frac{T_1}{2}$ wird in etwa eine sin² – Rampe gefahren.

Subindex	09_h
Kurzbeschreibung	Ruckbegrenzung (Verzögerung)(i.V.)
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	0..65535
Defaultwert	0

Beschreibung: Über diesen Index wird die Form der Bremsrampe vorgegeben. Der Wert muss kleiner als die Hälfte der Bremszeit (Subindex 07_h) gewählt werden. Die Ruckbegrenzung wirkt hier sinngemäß wie bei der Begrenzung für das Anfahren.

Subindex	$0A_h$
Kurzbeschreibung	Nummer des Folgefahrauftrages
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	0..180, 192..255
Defaultwert	0

Beschreibung: Über diesen Index wird die Nummer eines Folgefahrauftrags eingestellt. Ob mit diesem fortgefahren wird, wird über den Subindex 03_h, Bit 3 eingestellt.

Subindex	$0B_h$
Kurzbeschreibung	Startverzögerung Folgefahrauftrag
Einheit	ms
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	1..65535
Defaultwert	0

Beschreibung: Über dieses Objekt wird eine Verzögerungszeit eingestellt, mit der der Folgefahrauftrag gestartet wird. Dazu muss diese Funktion über Subindex 03_h, Bit 7 freigegeben werden.

Subindex	$0C_h$
Kurzbeschreibung	Kopieren eines Fahrauftrages
Einheit	—
Zugriff	w
PDO mapping	nicht möglich
Datentyp	2 x UNSIGNED16
Wertebereich	Jeweils 0..180, 192..255
Defaultwert	0, 0

Beschreibung: Über dieses Objekt können Fahraufträge kopiert werden.
Die Nummer, die zuerst im CAN – Telegramm erscheint, beschreibt dabei den Quell – Fahrauftrag, die folgende Nummer den Ziel – Fahrauftrag.

Subindex	$0D_h$
Kurzbeschreibung	Wichtungsfaktor Geschwindigkeit
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UNSIGNED16
Wertebereich	0..65535
Defaultwert	1

Beschreibung: Über dieses Objekt wird ein Multiplikator für die im RPDO Fahrsatz angegebene Geschwindigkeit eingestellt.

Subindex	$0E_h$
Kurzbeschreibung	Geschwindigkeit Direktfahrauftrag
Einheit	Inkremente / 250µs, bzw. Auflösungsabhängig
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über dieses Objekt wird die Geschwindigkeit für den Direktfahrauftrag (Fahrsatz 0) vorgegeben. Die Fahrauftragsart bestimmt dann, ob die Geschwindigkeit inkrementell oder als SI-Einheit ausgewertet wird.

IV.9 Latchfunktion

IV.9.1 SDO 2026h: Latchfreigabe

Index	2026_h
Kurzbeschreibung	Freigabe der Latchfunktion für CAN
Objektcode	RECORD
Number of elements	1

Beschreibung der Subindizes:

Subindex	01_h
Kurzbeschreibung	Freigabe der Latchfunktion für CAN
Einheit	—
Zugriff	r/w
PDO mapping	nicht möglich
Datentyp	UINTEGER8
Wertebereich	0, 1
Defaultwert	1

Beschreibung:

Der SERVOSTAR bietet die Möglichkeit über einen Latch - Eingang (Eingang 2, IN2MODE 26, auch konfigurierbar über SDO 3565_h, Subindex 01_h) eine Istposition hochgenau (Erfassungszeit < 1 µs) zu erfassen. Über das SDO 2026_h kann eingestellt werden, ob ein Latchimpuls über CAN gemeldet werden soll. Ein Wert von 0 bedeutet Sperren.

IV.10 Herstellerspezifische Istwerte

IV.10.1 SDO 2070h: Istwerte

Index	2070_h
Kurzbeschreibung	Istwerte
Objektcode	RECORD
Number of elements	16

Beschreibung:

Über diesen Index werden relevante Istwerte des SERVOSTAR zur Verfügung gestellt.

Beschreibung der Subindizes:

Subindex	01_h
Kurzbeschreibung	Istlage
Einheit	—
Zugriff	r
PDO mapping	möglich (vorgemappt auf wählbare TPDO 22, TPDO 32)
Datentyp	UNSIGNED32
Wertebereich	0..16777215
Defaultwert	0

Beschreibung: Über diesen Index kann die Motorposition innerhalb von 16 Umdrehungen eingelesen werden. Eine Umdrehung wird dabei mit einer Inkrementzahl von 20 Bit aufgelöst. Es gilt also:
1 Umdrehung $\Rightarrow 2^{20}$ Inkremente $\Rightarrow$1048576 Inkremente

Subindex	02_h
Kurzbeschreibung	Istdrehzahl
Einheit	min^{-1}
Zugriff	r
PDO mapping	möglich (vorgemappt auf wählbare TPDO 22, TPDO 32)
Datentyp	UNSIGNED32
Wertebereich	0..1677215
Defaultwert	0

Beschreibung: Über diesen Index kann die Motordrehzahl eingelesen werden.
Der Drehzahlistwert entspricht dabei:

$$n[min^{-1}] = 1875/262144 \times \text{eingelesener Drehzahlistwert}$$

Subindex	03_h
Kurzbeschreibung	Inkrementeller Positionsistwert
Einheit	—
Zugriff	r
PDO mapping	möglich (vorgemappt auf wählbares TPDO 33)
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann der Inkrementalwert der Istposition gelesen werden. Eine Umdrehung wird dabei mit einer Inkrementzahl von 20 Bit aufgelöst.
Es gilt also: 1 Umdrehung $\Rightarrow 2^{20}$ Inkremente = 1048576 Inkremente

Subindex	$04l_h$
Kurzbeschreibung	Lesen des 16 Bit Positionslatch
Einheit	—
Zugriff	r
PDO mapping	möglich
Datentyp	INTEGER16
Wertebereich	$-(2^{15})..(2^{15}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die gespeicherte (gelatchte) 16 Bit Position gelesen werden. Die Position wird in Inkrementen innerhalb einer Umdrehung ausgegeben. Die Ausgabe wird nicht von den Getriebefaktoren oder den Factor Groups beeinflusst.

Subindex	05_h
Kurzbeschreibung	Lesen des 32 Bit Positionslatch
Einheit	—
Zugriff	r
PDO mapping	möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31})..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die gespeicherte (gelatchte) 32 Bit Position gelesen werden. Die Position wird in Inkrementen innerhalb einer Umdrehung ausgegeben. Die Ausgabe wird nicht von den Getriebefaktoren oder den Factor Groups beeinflusst.

Subindex	06_h
Kurzbeschreibung	SI Positionsistwert
Einheit	µm
Zugriff	r
PDO mapping	möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die Istposition als SI – Einheit gelesen werden. Das Verhältnis von real zurückgelegter Strecke zu Motorumdrehungen wird durch $S_{SI} = S_{Inkr} \times PGEARI / PGEARO$ bestimmt, wobei PGEARO (SDO 2020_h, Subindex 08_h) die Anzahl der Inkremente enthält, die gefahren werden, wenn die zu fahrende Strecke PGEARI (SDO 2020_h, Subindex 09_h) beträgt. Zu beachten ist, dass dabei eine Motorumdrehung einer Inkrementzahl von 2^{20} = 1048576 entspricht.

Subindex	07_h
Kurzbeschreibung	SI Geschwindigkeitsistwert
Einheit	µm/s
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die Istgeschwindigkeit in SI – Einheiten gelesen werden.

Subindex	08_h
Kurzbeschreibung	Schleppfehler
Einheit	µm
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann der momentan gemessene Schleppfehler in SI – Einheiten bestimmt werden.

Subindex	09_h
Kurzbeschreibung	Effektivstrom
Einheit	mA
Zugriff	r
PDO mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	0..2 * Nennstrom [mA]
Defaultwert	0

Beschreibung: Über diesen Index kann der momentan gemessene Effektivstrom gelesen werden.

Subindex	$0A_h$
Kurzbeschreibung	Drehzahl
Einheit	min^{-1}
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die momentan gemessene Drehzahl gelesen werden.

Subindex	$0B_h$
Kurzbeschreibung	Kühlkörpertemperatur
Einheit	°C
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die Temperatur des Kühlkörpers bestimmt werden.

Subindex	$0C_h$
Kurzbeschreibung	Innentemperatur
Einheit	°C
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die Innentemperatur im Servoverstärker ausgelesen werden.

Subindex	$0D_h$
Kurzbeschreibung	Zwischenkreisspannung
Einheit	V
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die momentan gemessene Zwischenkreisspannung gelesen werden.

Subindex	$0E_h$
Kurzbeschreibung	Ballastleistung
Einheit	W
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die momentan gemessene Ballastleistung angezeigt werden.

Subindex	$0F_h$
Kurzbeschreibung	I^2T-Belastung
Einheit	%
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann die I^2t – Belastung gelesen werden.

Subindex	10_h
Kurzbeschreibung	Betriebsdauer
Einheit	min
Zugriff	r
PDO mapping	nicht möglich
Datentyp	INTEGER32
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	0

Beschreibung: Über diesen Index kann der Betriebsstundenzähler des Servoverstärkers ausgelesen werden.

Subindex	11_h
Kurzbeschreibung	erweiterter Status für TPDO 33
Einheit	—
Zugriff	r
PDO mapping	möglich
Datentyp	UNSIGNED32
Wertebereich	0..(2^{32}-1)
Defaultwert	—

Beschreibung:

Bit	Wert	Beschreibung
0	0x00000001	Status Eingang 1
1	0x00000002	Status Eingang 2
2	0x00000004	Status Eingang 3
3	0x00000008	Status Eingang 4
4	0x00000010	reserviert
5	0x00000020	reserviert
6	0x00000040	reserviert
7	0x00000080	reserviert
8	0x00000100	reserviert
9	0x00000200	reserviert
10	0x00000400	reserviert
11	0x00000800	reserviert
12	0x00001000	reserviert
13	0x00002000	reserviert
14	0x00004000	reserviert
15	0x00008000	reserviert
16	0x00010000	Auftrag aktiv (Lageregelung)
17	0x00020000	Referenzpunkt gesetzt
18	0x00040000	Home-Position
19	0x00080000	In-Position
20	0x00100000	Positionslatch erfolgt
21	0x00200000	frei
22	0x00400000	Meldung Position 1
23	0x00800000	Meldung Position 2
24	0x01000000	Meldung Position 3
25	0x02000000	Meldung Position 4
26	0x04000000	Initialisierung beendet
27	0x08000000	frei
28	0x10000000	Motorstillstand
29	0x20000000	Sicheheitsrelais
30	0x40000000	Endstufe freigegeben
31	0x80000000	Fehler steht an

IV.11 Profile Velocity Mode (pv) (DS402)

IV.11.1 Allgemeine Informationen

Der Profile Velocity Mode ermöglicht die Verarbeitung von Geschwindigkeitssollwerten und den zugehörigen Beschleunigungen.

IV.11.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$606C_h$	VAR	velocity_actual_value	INTEGER32	r
$60FF_h$	VAR	target_velocity	INTEGER32	r/w

IV.11.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040_h	VAR	controlword	INTEGER16	dc (⇒ IV.4)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ IV.4)
6063_h	VAR	position_actual_value*	INTEGER32	pc (⇒ IV.12)
6083_h	VAR	profile_acceleration	UNSIGNED32	pp (⇒ IV.14)
6084_h	VAR	profile_deceleration	UNSIGNED32	pp (⇒ IV.14)
6086_h	VAR	motion_profile_type	INTEGER16	pp (⇒ IV.14)
6094_h	ARRAY	velocity_encoder_factor	UNSIGNED32	fg (⇒ IV.5)

IV.11.4 Objektbeschreibung

IV.11.4.1 SDO 606Ch: velocity_actual_value* (DS402)

Das Objekt “velocity_actual_value” repräsentiert die aktuelle Drehzahl. Die Skalierung des Wertes hängt von dem Faktor “velocity_encoder_resolution” (SDO 6096_h) ab.

Index	$0x606C_h$
Name	velocity_actual_value
Objektcode	VAR
Datentyp	INTEGER32
Mode	pv
Zugriff	r
PDO Mapping	möglich
Einheit	velocity units
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

IV.11.4.2 SDO 60FFh: target_velocity (DS402)

Die Solldrehzahl (target_velocity) repräsentiert den Sollwert für den Rampengenerator. Die Skalierung des Wertes hängt von dem Faktor "velocity_encoder_resolution" (SDO 6096_h) ab.

Index	$0x60FF_h$
Name	target_velocity
Objektcode	VAR
Datentyp	INTEGER32
Mode	pv
Zugriff	r/w
PDO Mapping	möglich
Einheit	Inkremente
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

IV.12 Position Control Function (pc) (DS402)

IV.12.1 Allgemeine Informationen

In diesem Kapitel werden die Positionsistwerte beschrieben, die im Zusammenhang mit dem Lageregler des Antriebs stehen. Sie finden Verwendung im Profile Position Mode.

IV.12.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
6063_h	VAR	position_actual_value*	INTEGER32	r
6064_h	VAR	position_actual_value	INTEGER32	r

IV.12.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
$607A_h$	VAR	target_position	INTEGER32	pp (⇒ IV.14)
$607B_h$	VAR	position_range_limit	INTEGER32	pp (⇒ IV.14)
$607C_h$	VAR	home-offset	INTEGER32	hm (⇒ IV.13)
6093_h	VAR	position_factor	UNSIGNED32	fg (⇒ IV.5)
6094_h	ARRAY	velocity_encoder_factor	UNSIGNED32	fg (⇒ IV.5)
6096_h	ARRAY	acceleration_factor	UNSIGNED32	fg (⇒ IV.5)
6040_h	VAR	controlword	INTEGER16	dc (⇒ IV.4)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ IV.4)

IV.12.4 Objektbeschreibung

IV.12.4.1 SDO 6063h: position_actual_value* (DS402)

Das Objekt position_actual_value liefert die aktuelle Istposition in Inkremente. Die Auflösung pro Umdrehung kann 16 Bit oder 20 Bit betragen (s. Kommando PRBASE).

Index	$0x6063_h$
Name	position_actual_value
Objektcode	VAR
Datentyp	INTEGER32
Mode	pc, pp
Zugriff	r/w
PDO Mapping	möglich
Einheit	Inkremente (1 Umdr. = 16Bit / 20Bit)(s. PRBASE)f
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

IV.12.4.2 SDO 6064h: position_actual_value (DS402)

Das Objekt position_actual_value liefert die aktuelle Istposition. Die Auflösung (herstellerspez. Einheiten s. SDOs 2020_h 08_h/09_h oder nach Antriebsprofil DSP402 s. SDO $607A_h$) kann mit den Getriebefaktoren des Lageregler geändert werden.

Hinweis: Dieses Objekt sollte nicht in ein synchrones TPDO definiert (mapped) werden.

Index	$0x6064_h$
Name	position_actual_value
Objektcode	VAR
Datentyp	INTEGER32
Mode	pc, pp
Zugriff	r/w
PDO Mapping	möglich
Einheit	position units
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

IV.13 Homing Mode (hm) (DS402)

IV.13.1 Allgemeine Informationen

Dieses Kapitel beschreibt die verschiedenen Parameter, die zur Definition einer Referenzierung benötigt werden.

IV.13.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$607C_h$	VAR	home_offset	INTEGER32	r/w
6098_h	VAR	homing_method	INTEGER8	r/w
6099_h	ARRAY	homing_speeds	UNSIGNED32	r/w
$609A_h$	VAR	homing_acceleration	UNSIGNED32	r/w

IV.13.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040_h	VAR	controlword	INTEGER16	dc (⇒ IV.4)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ IV.4)

IV.13.4 Objektbeschreibung

IV.13.4.1 SDO 607Ch: home_offset (DS402)

Der Referenzoffset ist die Differenz zwischen der Nullposition der Anwendung und des Maschinennullpunktes. Alle nachfolgenden absoluten Fahraufträge berücksichtigen den Referenzoffset.

Index	$0x607C_h$
Name	home_offset
Objektcode	VAR
Datentyp	INTEGER32
Mode	hm
Zugriff	r/w
PDO Mapping	möglich
Einheit	benutzerdefiniert
Wertebereich	$(-2^{31})..(2^{31}-1)$
Defaultwert	0
EEPROM	ja

IV.13.4.2 SDO 6098h: homing_method (DS402)

Index	$0x6098_h$
Name	homing_method
Objektcode	VAR
Datentyp	INTEGER8
Mode	hm
Zugriff	r/w
PDO Mapping	möglich
Einheit	position units
Wertebereich	-128..127
Defaultwert	0
EEPROM	ja

Die folgenden Referenzfahrtmethoden werden unterstützt:

Methode nach DSP402	Kurzbeschreibung Referenz	ASCII-Kommando
-128..-4	reserviert	—
-3	Fahren auf mechanischen Anschlag, mit Nullpunktsuche	NREF = 7
-2	Setzen des Referenzpunktes auf die aktuelle Position unter Berücksichtigung des Schleppabstandes	NREF = 6
-1	Referenzfahrt innerhalb einer Umdrehung (Drehrichtung entfernungsabhängig)	NREF = 5, DREF= 2
0	reserviert	—
1	Referenzfahrt auf negativen Endschalter, mit Nullpunktsuche, Fahrtrichtung negativ	NREF = 2, DREF= 0
2	Referenzfahrt auf positiven Endschalter, mit Nullpunktsuche, Fahrtrichtung positiv	NREF = 2, DREF= 1
3..7	werden nicht unterstützt	—
8	Referenzfahrt mit Referenzschalter, mit Nullpunktsuche, Fahrtrichtung positiv	NREF = 1, DREF= 1
9..11	werden nicht unterstützt	—
12	Referenzfahrt mit Referenzschalter, mit Nullpunktsuche, Fahrtrichtung negativ	NREF = 1, DREF= 0
13..14	werden nicht unterstützt	—
15..16	reserviert	—
17	Referenzfahrt auf negativen Endschalter, ohne Nullpunktsuche, Fahrtrichtung negativ	NREF = 4, DREF= 0
18	Referenzfahrt auf negativen Endschalter, ohne Nullpunktsuche, Fahrtrichtung positiv	NREF = 4, DREF= 1
19..23	werden nicht unterstützt	—
24	Referenzfahrt auf Referenzschalter, ohne Nullpunktsuche, Fahrtrichtung positiv	NREF = 3, DREF= 1
25..27	werden nicht unterstützt	—
28	Referenzfahrt auf Referenzschalter, ohne Nullpunktsuche, Fahrtrichtung negativ	NREF = 3, DREF= 0
29..30	werden nicht unterstützt	—
31..32	reserviert	—
33	Referenzfahrt innerhalb einer Umdrehung, Fahrtrichtung negativ	NREF = 5, DREF= 0
34	Referenzfahrt innerhalb einer Umdrehung, Fahrtrichtung positiv	NREF = 5, DREF= 1
35	Setzen des Referenzpunktes an die aktuelle Position	NREF = 0
36..127	reserviert	—

IV.13.4.2.1 Beschreibung der Referenziermethoden

Durch die Auswahl einer Referenzfahrtart durch Beschreiben des Parameters homing_method (SDO 6098_h) wird folgendes bestimmt:

- das Referenzsignal (PStop, NStop, Referenzschalter)
- die Richtung der Referenzfahrt

und, wenn passend

- die Position des Nullimpulses

Die Referenzposition wird durch den Referenzoffset (SDO $607C_h$) festgelegt. Zur Anpassung der Grundeinstellung der Motorlage bei Referenzierung auf den Nullimpuls kann der herstellerspezifische Parameter ENCZERO (SDO 3537_h, Subindex 01_h) verwendet werden.

Eine ausführliche Beschreibung der Referenzfahrtarten finden Sie in der Beschreibung der Inbetriebnahmesoftware DRIVE.EXE.

IV.13.4.3 SDO 6099h: homing_speeds (DS402)

Index	$0x6099_h$
Name	homing_speeds
Objektcode	ARRAY
Number of elements	1
Datentyp	UNSIGNED32

Subindex	01_h
Kurzbeschreibung	speed_during_search_for_switch
Mode	hm
Zugriff	r/w
PDO Mapping	möglich
Einheit	velocity units
Wertebereich	0..(2^{32}-1)
EEPROM	ja
Defaultwert	2^{20}

IV.13.4.4 SDO 609Ah: homing_acceleration (DS402)

Index	$0x609A_h$
Name	homing_acceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	hm
Zugriff	r/w
PDO Mapping	möglich
Einheit	acceleration units
Wertebereich	0..(2^{32}-1)
Defaultwert	0
EEPROM	ja

IV.13.5 Homing Mode Sequence

Mit Setzen des Bit 4 (positive Flanke) wird die Referenzfahrt gestartet. Der erfolgreiche Abschluss wird mit Bit 12 im Zustandswort angezeigt (s. SDO 6041_h). Bit 13 zeigt einen Fehler an, der sich während der Referenzfahrt ereignet hat. Hier ist der Fehlercode auszuwerten:
Error register (SDOs 1001_h, 1003_h), manufacturer status (SDO1002_h)

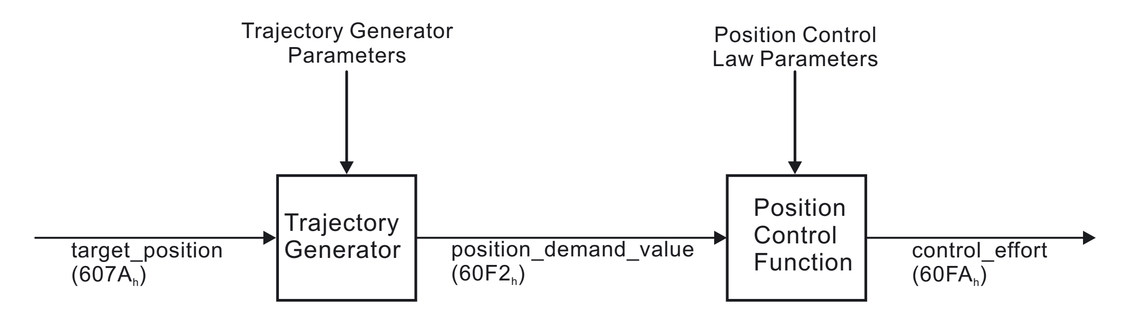
Bit 4	Bedeutung
0	Referenzfahrt inaktiv
0 ⇒ 1	Referenzfahrt starten
1	Referenzfahrt aktiv
0 ⇒ 1	Unterbrechung der Referenzfahrt

Bit 13	Bit 12	Bedeutung
0	0	Referenzpunkt nicht gesetzt bzw. Referenzfahrt noch nicht abgeschlossen
0	1	Referenzpunkt gesetzt bzw. Referenzfahrt erfolgreich abgeschlossen
1	0	Referenzfahrt konnte nicht erfolgreich abgeschlossen werden. (Schleppfehler)
1	1	kein erlaubter Zustand

IV.14 Profile Position Mode (pp)

IV.14.1 Allgemeine Informationen

Die Struktur dieser Betriebsart ist hier dargestellt:



Die spezielle Handshake-Verarbeitung von Controlword und Statusword wird in Kapitel IV.14.4.1 beschrieben.

IV.14.2 Objekte, die in diesem Kapitel definiert werden

Index	Objekt	Name	Type	Zugriff
$607A_h$	VAR	target_position	INTEGER32	r/w
$607B_h$	ARRAY	position_range_limit	INTEGER32	r/w
6081_h	VAR	profile_velocity	UNSIGNED32	r/w
6083_h	VAR	profile_acceleration	UNSIGNED32	r/w
6084_h	VAR	profile_deceleration	UNSIGNED32	r/w
6085_h	VAR	quick_stop_deceleration	UNSIGNED32	r/w
6086_h	VAR	motion_profile_type	INTEGER16	r/w

IV.14.3 Objekte, die in anderen Kapiteln definiert werden

Index	Objekt	Name	Type	Kapitel
6040_h	VAR	controlword	INTEGER16	dc (⇒ IV.4)
6041_h	VAR	statusword	UNSIGNED16	dc (⇒ IV.4)
$605A_h$	VAR	quick_stop_option_code	INTEGER16	dc (⇒ IV.4)
6093_h	ARRAY	position_factor	UNSIGNED32	fg (⇒ IV.5)
6094_h	ARRAY	velocity_encoder_factor	UNSIGNED32	fg (⇒ IV.5)
6097_h	ARRAY	acceleration_factor	UNSIGNED32	fg (⇒ IV.5)

IV.14.4 Objektbeschreibung

IV.14.4.1 SDO 607Ah: target_position (DS402)

Das Objekt target_position definiert die Zielposition des Antriebes. Abhängig vom Bit 6 im controlword wird die Zielposition als relativer Weg oder als absolute Position interpretiert. Dabei kann die Art der Relativfahrt durch den herstellerspezifischen Parameter 2022_h Subindex 03_h weiter aufgeschlüsselt werden.
Die mechanische Auflösung wird über die Getriebefaktoren SDO 6093_h Subindex 01_h und 02_h eingestellt.

Index	$0x607A_h$
Name	target_position
Objektcode	VAR
Datentyp	INTEGER32
Mode	pp
Zugriff	r/w
PDO Mapping	möglich
Einheit	benutzerdefiniert
Wertebereich	$-(2^{31}-1)..(2^{31}-1)$
Defaultwert	—
EEPROM	nein

IV.14.4.2 SDO 607Bh: position_range_limit (DS402)

Mit dem Objekt “position_range_limit” wird der Anfang und das Ende des Verfahrbereiches für eine Modulo - Achse festgelegt. Der Anfang des Bereiches wird mit Subindex 01_h “min_position_range_limit” (ASCII SRND) und das Ende mit Subindex 02_h “max_position_range_limit” (ASCII ERND) definiert. Diese Funktionalität kann erst nach einer Neukonfiguration des Geräts verwendet werden. Dazu muss das SDO 2020_h Subindex 01_h mit dem Wert 2 versehen werden und danach die Konfigurationsprozedur gestartet werden (⇒ Kap. VI.4).

Index	$0x607B_h$
Name	position_range_limit
Objektcode	ARRAY
Number of elements	2
Datentyp	INTEGER32

Subindex	01_h
Kurzbeschreibung	min_position_range_limit
Mode	pp, pc
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	position units
Wertebereich	$(-2^{31})..(2^{31}-1)$
EEPROM	ja
Defaultwert	-2^{31}

Subindex	02_h
Kurzbeschreibung	max_position_range_limit
Mode	pp, pc
Zugriff	r/w
PDO Mapping	nicht möglich
Einheit	position units
Wertebereich	$(-2^{31})..(2^{31}-1)$
EEPROM	ja
Defaultwert	$2^{31}-1$

IV.14.4.3 SDO 6081h: profile_velocity (DS402)

Die profile_velocity ist die Endgeschwindigkeit, die nach der Beschleunigungsphase eines Fahrauftrages erreicht werden soll. Die verwendete Skalierung ist abhängig von dem eingestellten “velocity_encoder_factor” (SDO 6094_h).
Der Sollwert wird in Abhängigkeit des eingestellten Operationsmode (pp, pv) benutzt.

Index	$0x6081_h$
Name	profile_velocity
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp, pv
Zugriff	r/w
PDO Mapping	möglich
Einheit	speed units
Wertebereich	$0..(2^{32}-1)$
Defaultwert	10
EEPROM	nein

IV.14.4.4 SDO 6083h: profile_acceleration (DS402)

Die Beschleunigungsrampe (profile_acceleration) wir in Einheiten, die der Anwender definiert hat, vorgegeben. Die Verarbeitung bzw. Interpretation des Beschleunigungssollwertes kann auf zwei Arten erfolgen:

- **Interne Getriebefaktoren PGEARI = PGEARO (s. SDOs 2020_h 08_h/09_h, s. ASCII - Kommando PGEARI/PGEARO)**
  Die Beschleunigungsrampe wird als Beschleunigungszeit [ms] oder Beschleinigungssteigung [Inkr./s^2] bezogen auf die Zielgeschwindigkeit (SDO 6081_h "profile_velocity") interpretiert. Die Skalierung des Wertes ist abhängig vom Beschleunigungsfaktor (SDO 6097_h "acceleration_factor").
  Hinweis: Z. Zt. wird nur die Einheit Beschleunigungszeit unterstützt!

- **Interne Getriebefaktoren PGEARI <> PGEARO (s. SDOs 2020_h 08_h/09_h, s. ASCII - Kommando PGEARI/PGEARO)**
  Die Beschleunigungsrampe wird als Beschleunigungszeit [ms] oder Beschleinigungssteigung [Längeneinheit/s^2] bezogen auf die Zielgeschwindigkeit interpretiert. Die Skalierung des Wertes ist abhängig von den eingestellten Getriebefaktoren (s. Beschreibung der ASCII –Kommandos PGEARI und PGEARO) und der eingestellten Basiseinheiten [ms] oder [Längeneinheit/s^2]. Die Vorwahl der Basiseinheit erfolgt über Bit 12 im controlword des Fahrauftrages (SDO 2022_h03_h, ASCII – Kommando O_C).

Die Art der Beschleunigungsrampe kann als lineare Rampe oder als Sin^2 Rampe ausgewählt werden (s. SDO 6086_h).

Index	$0x6083_h$
Name	profile_acceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	r/w
PDO Mapping	möglich
Einheit	acceleration units
Wertebereich	0..(2^{32}-1)
Defaultwert	0

IV.14.4.5 SDO 6084h: profile_deceleration (DS402)

Die Bremsrampe wird analog zur Beschleunigungsrampe behandelt. (s. SDO 6083_h)

Index	$0x6084_h$
Name	profile_deceleration
Objektcode	VAR
Datentyp	UNSIGNED32
Mode	pp
Zugriff	r/w
PDO Mapping	möglich
Einheit	acceleration units
Wertebereich	0..(2^{32}-1)
Defaultwert	0

IV.14.4.6 SDO 6086h: motion_profile_type (DS402)

Die Art der Beschleunigungsrampe kann mit diesem Objekt als lineare Rampe oder als Sin² Rampe ausgewählt werden.

Index	0x6086$_h$
Name	motion_profile_type
Objektcode	VAR
Datentyp	INTEGER16
Mode	pp
Zugriff	r/w
PDO Mapping	möglich
Einheit	keine
Wertebereich	$(-2^{15})..(2^{15}-1)$
Defaultwert	—
EEPROM	ja

profile code	profile type
-32768..-1	herstellerspez. (wird nicht unterstützt)
0	linear (trapez)
1	sin²
2..32767	profilspez. Erweiterungen (wird nicht unterstützt)

IV.14.5 Funktionelle Beschreibung

In diesem Profil werden zwei Wege der Positionssollwertübergabe an den Antrieb unterstützt.

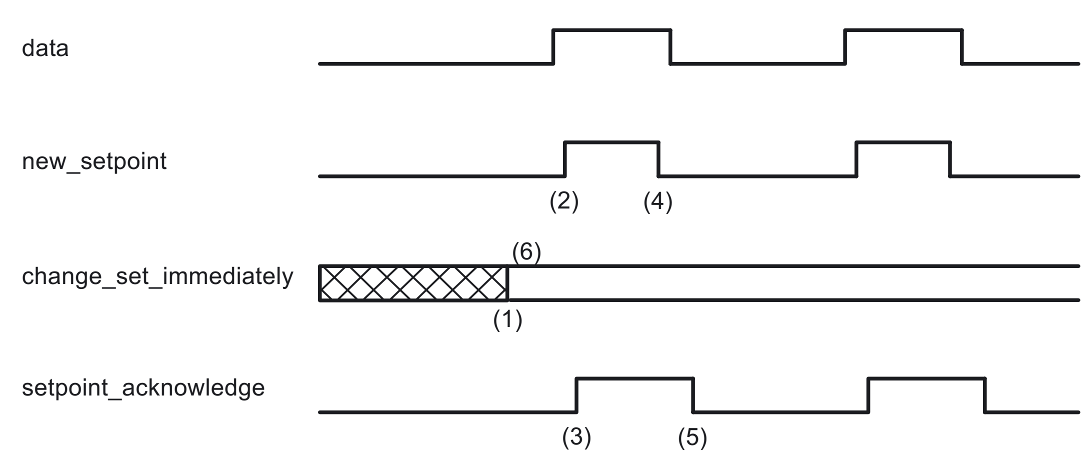
Eine Folge von Sollwerten:

Nach Erreichen der Zielposition berechnet der Antrieb sofort die Bewegung zur vorher übergebenen neuen Zielposition. Dies führt zu einer kontinuierlichen Bewegung, ohne dass der Antrieb zwischendurch auf Geschwindigkeit 0 abbremst. Beim SERVOSTAR ist dies nur bei Verwendung von Trapezrampen möglich.

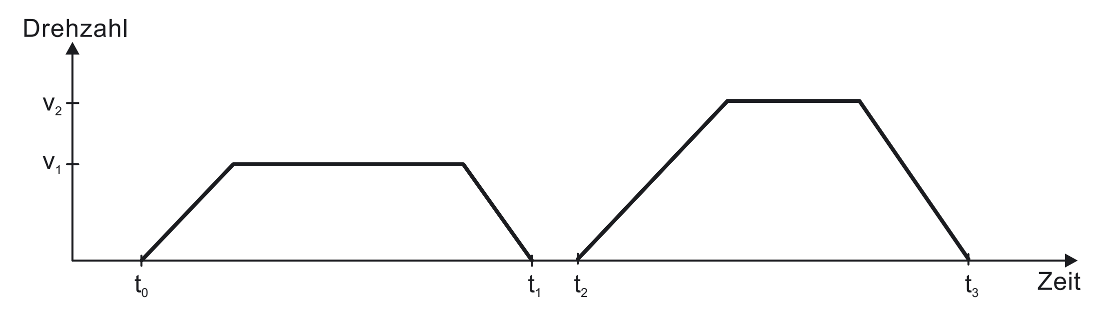
Einzelne Sollwerte:

Nach Erreichen der Zielposition signalisiert der Antrieb an den Master, dass er das Ziel erreicht hat und erhält dann einen neuen Sollwert. Nach Erreichen der Zielposition ist die Geschwindigkeit normalerweise 0 bevor die Achse zu einer neuen Zielposition fährt.

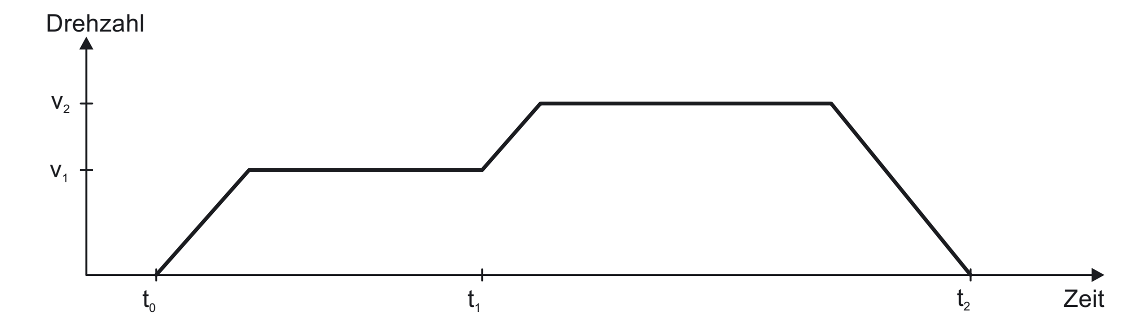
Die zwei Arten werden über das Timing der Bits new_setpoint, change_set_immediately des Steuerworts und das Bit setpoint_acknowledge des Zustandswort gesteuert. Diese Bits ermöglichen einen Handshake - Mechanismus, der es ermöglicht einen neuen Sollwert vorzubereiten während ein alter Fahrauftrag ausgeführt wird. Das verringert Reaktionszeiten innerhalb eines Steuerungsprogramms in einer Steuerung.



Die Bilder zeigen den Unterschied zwischen einer "Folge von Sollwerten" und einzelnen Sollwerten. Der Startwert des Bits change_set_immediatly im Steuerwort entscheidet über die verwendete Art. Die betrachteten Beispiele gelten nur für Trapezrampen. Wenn das Bit change_set_immediatly auf 0 ist (durchgehender Strich in Bild 1) wird ein einzelner Sollwert vom Antrieb erwartet (1). Nachdem Daten an den Antrieb übergeben wurden, signalisiert der Master durch den Wechsel am Bit new_setpoint im Steuerwort auf "1", dass die Daten gültig sind (2). Der Antrieb antwortet mit dem setpoint_acknowledge Bit = 1 im Zustandswort nachdem er den Wert erkannt und gespeichert hat (3). Nun kann der Master das Bit new_setpoint auf 0 setzen (4) woraufhin der Antrieb durch Rücksetzen des Bits setpoint_acknowledge signalisiert, dass er wieder neue Sollwerte entgegennehmen kann (5). In Bild 2 führt dieser Mechanismus zu einer Geschwindigkeit von 0 nachdem eine Rampe gefahren wurde, um eine Zielposition X_1 zur Zeit t_1 zu erreichen. Nach dem Signal an den Master, dass das Ziel erreicht wurde, wird die neue Zielposition zum Zeitpunkt t_2 verarbeitet und zum Zeitpunkt t_3 erreicht.



Mit dem Bit change_set_immediatly = 1 (symbolisiert durch die gestrichelte Linie in Bild 1) weist der Master den Antrieb an direkt nach dem Erreichen des letzten Sollwerts einen neuen Sollwert zu verarbeiten. Das Timing der Signale bleibt das gleiche. Diese Vorgehensweise bewirkt beim Antrieb, dass er schon den nächsten Sollwert X_2 verarbeitet und Geschwindigkeit behält, wenn er die Zielposition X_1 zum Zeitpunkt t_1 erreicht. Danach fährt der Antrieb zur schon berechneten Zielposition X_2.



Bits im controlword:
Bit 4 new_set_point (positive Flanke!)
Bit 5 change_set_immediatly
Bit 6 absolut / relativ

Bits im statusword:
Bit 12 setpoint acknowledge
Bit 13 following error

Hinweis zur Fahrauftragsart “relativ”:

Wird das Bit 6 gesetzt, ist die Fahrauftragsart “relativ in Abhängigkeit zur letzten Zielposition oder Istposition” aktiviert. Sollten andere Relativarten gewünscht sein, sind diese im Vorfeld mit dem SDO 2022_h Subindex 03_h “Positionierdaten für den Mode Lage” zu aktivieren (s. auch SDO 2022_h Subindex 03_h oder ASCII Objekt O_C).

Hinweis zum Mode Profile Position Mode:

Funktionale Beschreibung für den Mode: Profile Position Mode
Das Antriebsprofil DSP402 unterscheidet zwei Methoden Zielpositionen anzufahren. Diese beiden Methoden werden über die Bits “new_setpoint” und “change_set_immediatly” im controlword und “setpoint_acknowledge” im statusword gesteuert. Mit Hilfe dieser Bits kann ein Fahrauftrag aufgebaut werden, während ein anderer gerade ausgeführt wird (Handshake).

- **Anfahren von mehreren Zielpositionen ohne Zwischenstopp**
  Nachdem die Zielposition erreicht wurde, wird sofort die nächste Zielposition angefahren. Voraussetzung ist, dass dem Antrieb neue Sollwerte signalisiert werden. Dies wird mit einer positiven Flanke (Bit “new_setpoint”) realisiert. Das Bit “setpoint_acknowledge” darf hierbei nicht gesetzt sein (siehe auch Handshake DSP402).
  Die Geschwindigkeit wird, nach erreichen des ersten Zielpunktes, nicht auf Null reduziert.

- **Anfahren von einer einzelnen Zielpositionen**
  Der Antrieb fährt in die Zielposition, wobei die Geschwindigkeit auf Null reduziert wird. Das erreichen der Zielposition wird mit dem Bit “target_reached” im statusword signalisiert.

V Der Objektkanal

V.1 Objektbeschreibung

V.1.1 SDO > 3500h: Herstellerspezifischer Objektkanal

Das Objektverzeichnis ist ab Index 3500_h (reservierter Objektbereich 3500_h– 3900_h) um sämtliche Geräteobjekte erweitert worden, die mit bis zu 4 Byte Nutzdaten darstellbar sind. Dieser Bereich ist dynamisch erweiterbar d.h., neue Geräteparameter, die das o. g. Datenformat erfüllen, werden bei Erweiterungen in der “Kernfirmware” **automatisch** der Tabelle angehängt. Mit Hilfe des SDO 3500_h (Subindex 01_h, read) kann die Gesamtanzahl der Objekte im Objektkanal bestimmt werden (⇒ Kap. VI.3).

Jedes Objekt in diesem Bereich wird mit Hilfe von 8 Subindizes beschrieben. Diese Struktur baut sich folgendermaßen auf:

Index	> 3500_h
Name	Objektabhängig
Objektcode	VAR
Datentyp	RECORD

Subindex	00_h
Beschreibung	Anzahl der Einträge
Einheit	—
Zugriff	—
PDO Mapping	nicht möglich
Datentyp	UNSIGNED8
Wertebereich	$0...2^8$-1
EEPROM	—
Defaultwert	—

Subindex	01_h
Beschreibung	Lesen / Schreiben eines Parameters
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	s. jeweiliges ASCII-Kommando
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	s. Subindex 04_h
Defaultwert	s. jeweiliges ASCII-Kommando

Subindex	02_h
Beschreibung	Lesen des unteren Grenzwertes
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	r
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Subindex	03_h
Beschreibung	Lesen des oberen Grenzwertes
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	r
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Subindex	04_h
Beschreibung	Lesen des Defaultwertes
Einheit	s. jeweiliges ASCII-Kommando
Zugriff	r
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Subindex	05_h
Beschreibung	Lesen des Parameterformats
Einheit	—
Zugriff	r
PDO Mapping	nicht möglich
Datentyp	s. jeweiliges ASCII-Kommando
Wertebereich	s. jeweiliges ASCII-Kommando
EEPROM	—
Defaultwert	—

Beschreibung:

Die folgenden Parameterformate sind möglich:

0	Funktion (keine Parameter)
1	Funktion (INTEGER32 Parameter)
2	Funktion (INTEGER32 Parameter mit Wichtung 3)
3	INTEGER8
4	UINTEGER8
5	INTEGER16
6,13	UINTEGER16
7	INTEGER32
8,12	UINTEGER32
9,10	INTEGER32 (Wichtung 3)



Achtung:
Auf Parameter mit dem Parameterformat 0 darf nur lesend zugegriffen werden!

Subindex	06_h
Beschreibung	Lesen der Parameter-Kontrolldaten
Einheit	—
Zugriff	r
PDO Mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	$0..2^{32}-1$
EEPROM	—
Defaultwert	—

Beschreibung:

0x00010000 Nach einer Änderung muss die Variable abgespeichert, und der Regler resettiert werden.

0x00020000 Variable wird im seriellen EEPROM abgespeichert.

0x00200000 Variable ist read-only, darf nicht über den Bus geschrieben werden.

Subindex	07_h / 08_h
Beschreibung	reserviert
Einheit	—
Zugriff	r
PDO Mapping	nicht möglich
Datentyp	UNSIGNED32
Wertebereich	$0..2^{32}-1$
EEPROM	—
Defaultwert	—

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3500	MAXSDO	INTEGER32	—	—	—
3501	ACC	INTEGER16	—	—	Ja
3502	ACCR	INTEGER16	—	—	Ja
3503	ACTFAULT	INTEGER8	—	Disabled + Reset	Ja
3504	ACTIVE	INTEGER8	—	—	Nein
3505	ADDR	UNSIGNED8	—	—	Ja
3506	AENA	INTEGER8	—	—	Ja
3507	ANCNFG	INTEGER8	—	Disabled + Reset	Ja
3508	ANDB	INTEGER32	☒	—	Ja
3509	ANIN1	INTEGER32	—	—	Nein
350A	ANIN2	INTEGER32	—	—	Nein
350B	ANOFF1	INTEGER16	—	—	Ja
350C	ANOFF2	INTEGER16	—	—	Ja
350D	ANOUT1	INTEGER8	—	Disabled + Reset	Ja
350E	ANOUT2	INTEGER8	—	Disabled + Reset	Ja
350F	ANZERO1	—	—	—	—
3510	ANZERO2	—	—	—	—
3511	AVZ1	INTEGER32	☒	—	Ja
3512	CALCHP	—	—	Enabled	—
3513	CALCRK	—	—	Enabled	—
3514	CALCRP	—	—	Disabled + Reset	—
3515	CBAUD	INTEGER16	—	Disabled + Reset	Ja
3516	reserviert				
3517	CDUMP	—	—	—	—
3518	CLRFAULT	—	—	—	—
3519	CLRHR	—	—	—	—
351A	CLRORDER	INTEGER16	—	Disabled	—
351B	CLRWARN	UNSIGNED8	—	Disabled + Reset	Ja
351C	CONFIG	—	—	—	—
351D	CONTINUE	—	—	Enabled	—
351E	CTUNE	—	—	Enabled	—
351F	CUPDATE	—	—	Disabled	—

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3520	DAOFFSET1	INTEGER16	—	—	Ja
3521	DAOFFSET2	INTEGER16	—	—	Ja
3522	DEC	INTEGER16	—	—	Ja
3523	DECDIS	INTEGER16	—	—	Ja
3524	DECR	INTEGER16	—	—	Ja
3525	DECSTOP	INTEGER16	—	—	Ja
3526	DEVICE	—	—	—	Nein
3527	DICONT	INTEGER32	☒	—	Nein
3528	DIFVAR	—	—	—	—
3529	DIPEAK	INTEGER32	☒	—	Nein
352A	DIR	INTEGER8	—	Disabled + Reset	Ja
352B	DIS	—	—	Enabled	—
352C	DREF	INTEGER8	—	—	Ja
352D	DRVSTAT	INTEGER32	—	—	Nein
352E	DR_TYPE	INTEGER16	—	—	Nein
352F	DUMP	—	—	—	—
3530	EN	—	—	Disabled	—
3531	ENCCAPT	INTEGER8	—	Disabled	Ja
3532	ENCIN	INTEGER32	—	Disabled + Reset	Ja
3533	ENCLINES	INTEGER16	—	Disabled + Reset	Ja
3534	ENCMODE	INTEGER8	—	—	Ja
3535	ENCOUT	INTEGER16	—	—	Ja
3536	reserviert				
3537	ENCZERO	INTEGER16	—	—	Ja
3538	EXTMUL	INTEGER16	—	—	Ja
3539	EXTPOS	INTEGER8	—	Disabled + Reset	Ja
353A	EXTWD	INTEGER32	—	—	Ja
353B	FBTYPE	INTEGER8	—	Disabled + Reset	Ja
353C	FILTMODE	UNSIGNED8	—	Disabled + Reset	Ja
353D	FOLDMODE	INTEGER8	—	Disabled + Reset	Ja
353E	GEARI	INTEGER16	—	—	Ja
353F	GEARMODE	INTEGER8	—	Disabled + Reset	Ja
3540	GEARO	INTEGER16	—	—	Ja
3541	GET	—	—	—	—
3542	GP	INTEGER32	☒	—	Ja
3543	GPFBT	INTEGER32	☒	—	Ja
3544	GPFFT	INTEGER32	☒	—	Ja
3545	GPFFV	INTEGER32	☒	—	Ja
3546	GPTN	INTEGER32	☒	—	Ja
3547	GPV	INTEGER32	☒	—	Ja
3548	GV	INTEGER32	☒	—	Ja
3549	GVFBT	INTEGER32	☒	—	Ja
354A	GVFILT	INTEGER8	—	—	Ja
354B	GVFR	INTEGER32	☒	—	Ja
354C	GVT2	INTEGER32	☒	—	Ja
354D	GVTN	INTEGER32	☒	—	Ja
354E	HACOFFS	INTEGER16	—	—	Encoder
354F	HAFACT1	INTEGER16	—	—	Encoder
3550	HASOFFS	INTEGER16	—	—	Encoder
3551	HDUMP	—	—	—	—
3552	HICOFFS	INTEGER16	—	—	Ja
3553	HIFACT1	INTEGER16	—	—	Encoder
3554	HISOFFS	INTEGER16	—	—	Encoder
3555	HRESET	—	—	—	—
3556	HSAVE	—	—	—	—
3557	HVER	—	—	—	Nein
3558	I	INTEGER32	☒	—	Nein
3559	I2T	INTEGER32	—	—	Nein
355A	I2TLIM	INTEGER8	—	—	Ja
355B	ICMD	INTEGER32	☒	—	Nein

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
355C	ICONT	INTEGER32	⊠	—	Ja
355D	ID	INTEGER32	⊠	—	Nein
355E	IDUMP	—	—	—	—
355F	IMAX	INTEGER32	⊠	—	Nein
3560	IN	—	—	—	—
3561	IN1	INTEGER8	—	—	—
3562	IN1MODE	INTEGER8	—	Disabled + Reset	Ja
3563	IN1TRIG	INTEGER32	—	—	Ja
3564	IN2	INTEGER8	—	—	Nein
3565	IN2MODE	INTEGER8	—	Disabled + Reset	Ja
3566	IN2TRIG	INTEGER32	—	—	Ja
3567	IN3	INTEGER8	—	—	Nein
3568	IN3MODE	INTEGER8	—	Disabled + Reset	Ja
3569	IN3TRIG	INTEGER32	—	—	Ja
356A	IN4	INTEGER8	—	—	Nein
356B	IN4MODE	INTEGER8	—	Disabled + Reset	Ja
356C	IN4TRIG	INTEGER32	—	—	Ja
356D	INPOS	—	—	—	—
356E	IPEAK	INTEGER32	⊠	—	Ja
356F	IPEAKN	INTEGER32	⊠	—	Ja
3570	IQ	INTEGER32	⊠	—	Nein
3571	ISCALE1	INTEGER32	⊠	—	Ja
3572	ISCALE2	INTEGER32	⊠	—	Ja
3573	K	—	—	Enabled	Nein
3574	KC	INTEGER32	⊠	—	Ja
3575	KEYLOCK	INTEGER8	—	—	Ja
3576	reserviert				
3577	L	INTEGER32	⊠	—	Ja
3578	LATCH16	INTEGER16	—	—	—
3579	LATCH16N	INTEGER16	—	—	Nein
357A	LATCH32	INTEGER32	—	—	—
357B	LATCH32N	INTEGER32	—	—	Nein
357C	LATCHX32	INTEGER32	—	—	Nein
357D	LATCHX32N	INTEGER32	—	—	Nein
357E	LED1	INTEGER8	—	—	—
357F	LED2	INTEGER8	—	—	—
3580	LED3	INTEGER8	—	—	—
3581	LEDSTAT	INTEGER16	—	—	—
3582	LIST	—	—	—	Nein
3583	LOAD	—	—	—	Nein
3584	MAXTEMPE	INTEGER16	—	—	Ja
3585	MAXTEMPH	INTEGER16	—	—	Ja
3586	MAXTEMPM	INTEGER32	⊠	—	Ja
3587	MBRAKE	INTEGER8	—	Disabled + Reset	Ja
3588	MDBCNT	—	—	—	—
3589	MDBGET	—	—	—	—
358A	MDBSET	INTEGER16	—	—	—
358B	MDUMP	—	—	—	—
358C	reserviert				
358D	MH	—	—	Enabled	—
358E	MICONT	INTEGER32	⊠	—	Ja
358F	MIPEAK	INTEGER32	⊠	—	Ja
3590	reserviert				
3591	MJOG	—	—	Enabled	—
3592	MVANGLP	INTEGER16	—	—	Ja
3593	MKT	INTEGER32	⊠	—	Ja
3594	reserviert				
3595	MLGC	INTEGER32	⊠	—	Ja
3596	MLGD	INTEGER32	⊠	—	Ja
3597	MLGP	INTEGER32	⊠	—	Ja

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3598	MLGQ	INTEGER32	☒	—	Ja
3599	MNUMBER	INTEGER16	—	Disabled	Ja
359A	MONITOR1	INTEGER16	—	—	Nein
359B	MONITOR2	INTEGER16	—	—	Nein
359C	MPHASE	INTEGER16	—	Disabled	Ja
359D	MPOLES	INTEGER8	—	Disabled	Ja
359E	MRD	—	—	Enabled	—
359F	reserviert				
35A0	MRESBW	INTEGER16	—	—	Ja
35A1	MRESPOLES	INTEGER8	—	Disabled	Ja
35A2	MSG	INTEGER8	—	—	Ja
35A3	MSPEED	INTEGER32	☒	—	Ja
35A4	reserviert				
35A5	MTANGLP	INTEGER16	—	—	Ja
35A6	reserviert				
35A7	MVANGLB	INTEGER32	—	—	Ja
35A8	MVANGLF	INTEGER16	—	—	Ja
35A9	M_RESET	—	—	Disabled	—
35AA	NONBTB	INTEGER8	—	—	Ja
35AB	NOTCHBW	INTEGER32	☒	—	Ja
35AC	NOTCHHZ	INTEGER32	☒	—	Ja
35AD	NREF	INTEGER8	—	—	Ja
35AE	O1	INTEGER8	—	—	Nein
35AF	O1MODE	INTEGER8	—	Disabled + Reset	Ja
35B0	O1TRIG	INTEGER32	—	—	Ja
35B1	O2	INTEGER8	—	—	Nein
35B2	O2MODE	INTEGER8	—	Disabled + Reset	Ja
35B3	O2TRIG	INTEGER32	—	—	Ja
35B4	OPMODE	INTEGER8	—	—	Ja
35B5	OPTION	INTEGER16	—	—	Nein
35B6	OVRIDE	INTEGER8	—	—	Ja
35B7	O_ACC1	INTEGER16	—	—	Nein
35B8	O_ACC2	INTEGER16	—	—	Nein
35B9	O_C	INTEGER16	—	—	Nein
35BA	O_DEC1	INTEGER16	—	—	Nein
35BB	O_DEC2	INTEGER16	—	—	Nein
35BC	O_FN	INTEGER16	—	—	Nein
35BD	O_FT	INTEGER16	—	—	Nein
35BE	O_P	INTEGER32	—	—	Nein
35BF	O_V	INTEGER32	—	—	Nein
35C0	PBAL	INTEGER32	—	—	Nein
35C1	PBALMAX	INTEGER32	—	—	Ja
35C2	PBALRES	INTEGER8	—	—	Ja
35C3	PBAUD	INTEGER32	☒	—	Nein
35C4	PDUMP	—	—	—	—
35C5	PE	INTEGER32	—	—	Nein
35C6	PEINPOS	INTEGER32	—	—	Ja
35C7	PEMAX	INTEGER32	—	—	Ja
35C8	PFB	INTEGER32	—	—	Nein
35C9	PFB0	INTEGER32	—	—	Nein
35CA	PGEARI	INTEGER32	—	Disabled + Reset	Ja
35CB	PGEARO	INTEGER32	—	Disabled + Reset	Ja
35CC	PIOBUF	—	—	—	Nein
35CD	PMODE	INTEGER32	—	Disabled + Reset	Ja
35CE	PNOID	INTEGER32	—	—	Nein
35CF	POSCNFG	INTEGER8	—	Disabled + Reset	Ja
35D0	PPOTYP	INTEGER8	—	—	Ja
35D1	PRBASE	INTEGER8	—	Disabled + Reset	Ja
35D2	PRD	INTEGER32	—	—	Nein
35D3	PROMPT	INTEGER16	—	—	Nein
35D4	PSTATE	—	—	—	Nein

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
35D5	PTBASE	INTEGER8	—	—	Ja
35D6	PTMIN	INTEGER16	—	—	Ja
35D7	PV	INTEGER32	—	—	Nein
35D8	PVMAX	INTEGER32	—	—	Ja
35D9	PVMAXN	INTEGER32	—	—	Ja
35DA	reserviert				
35DB	reserviert				
35DC	READNIMP	—	—	—	—
35DD	READY	INTEGER8	—	—	Nein
35DE	RECDONE	INTEGER8	—	—	Nein
35DF	RECING	INTEGER8	—	—	Nein
35E0	RECOFF	—	—	—	—
35E1	RECRDY	INTEGER8	—	—	Nein
35E2	REFIP	INTEGER32	⊠	—	Ja
35E3	REFPOS	INTEGER32	—	—	Nein
35E4	REMOTE	INTEGER8	—	—	Nein
35E5	RESPHASE	INTEGER16	—	—	Ja
35E6	RK	INTEGER16	—	—	Ja
35E7	ROFFS	INTEGER32	—	—	Ja
35E8	RS232T	INTEGER16	—	—	Ja
35E9	RSTVAR	—	—	Disabled	Nein
35EA	S	—	—	—	—
35EB	SAVE	—	—	—	—
35EC	SBAUD	INTEGER8	—	—	Ja
35ED	SCAN	—	—	—	—
35EE	SDUMP	—	—	—	—
35EF	SERIALNO	INTEGER32	—	—	Nein
35F0	SETREF	—	—	—	—
35F1	SETROFFS	—	—	—	—
35F2	SLEN	INTEGER8	—	—	Ja
35F3	SLOTIO	INTEGER32	—	—	Nein
35F4	SPHAS	INTEGER8	—	—	Nein
35F5	SPSET	INTEGER16	—	—	Ja
35F6	SSIGRAY	INTEGER8	—	Disabled	Ja
35F7	SSIINV	INTEGER8	—	Disabled	Ja
35F8	SSIMODE	INTEGER8	—	—	Ja
35F9	SSIOUT	INTEGER8	—	Disabled	Ja
35FA	SSTAT	—	—	—	Nein
35FB	STAT	INTEGER16	—	—	Nein
35FC	STATIO	—	—	—	Nein
35FD	STATUS	—	—	—	Nein
35FE	STOP	—	—	Enabled	—
35FF	STOPMODE	INTEGER8	—	Disabled + Reset	Ja
3600	SWCNFG	UNSIGNED16	—	Disabled + Reset	Ja
3601	SWCNFG2	UNSIGNED16	—	Disabled + Reset	Ja
3602	SWE0	INTEGER32	—	—	Ja
3603	SWE0N	INTEGER32	—	—	Ja
3604	SWE1	INTEGER32	—	—	Ja
3605	SWE1N	INTEGER32	—	—	Ja
3606	SWE2	INTEGER32	—	—	Ja
3607	SWE2N	INTEGER32	—	—	Ja
3608	SWE3	INTEGER32	—	—	Ja
3609	SWE3N	INTEGER32	—	—	Ja
360A	SWE4	INTEGER32	—	—	Ja
360B	SWE4N	INTEGER32	—	—	Ja
360C	SWE5	INTEGER32	—	—	Ja
360D	SWE5N	INTEGER32	—	—	Ja
360E	T	INTEGER32	⊠	Enabled	—
360F	TASK	—	—	—	Nein
3610	TEMPE	INTEGER32	—	—	Nein
3611	TEMPH	INTEGER32	—	—	Nein

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
3612	TEMPM	INTEGER32	—	—	Nein
3613	TRJSTAT	INTEGER32	—	—	Nein
3614	TRUN	—	—	—	Ja
3615	reserviert				
3616	UID	INTEGER16	—	—	Ja
3617	UVLTMODE	INTEGER8	—	Disabled + Reset	Ja
3618	V	INTEGER32	—	—	Nein
3619	reserviert				
361A	VBUS	INTEGER32	—	—	Nein
361B	VBUSBAL	INTEGER16	—	—	Ja
361C	VBUSMAX	INTEGER32	—	—	Ja
361D	VBUSMIN	INTEGER16	—	—	Ja
361E	VCMD	INTEGER32	⊠	—	Nein
361F	VDUMP	—	—	—	—
3620	VELO	INTEGER32	⊠		Ja
3621	VJOG	INTEGER32	—	—	Ja
3622	VLIM	INTEGER32	⊠	—	Ja
3623	VLIMN	INTEGER32	⊠	—	Ja
3624	VMAX	INTEGER32	⊠	—	Nein
3625	VMIX	INTEGER32	⊠	—	Ja
3626	VMUL	INTEGER32	—	—	Ja
3627	VOSPD	INTEGER32	⊠	—	Ja
3628	VREF	INTEGER32	—	—	Ja
3629	VSCALE1	INTEGER16	—	—	Ja
362A	VSCALE2	INTEGER16	—	—	Ja
362B	\	UNSIGNED8	—	—	—
362C	DILIM	INTEGER8	—	Disabled + Reset	Ja
362D	DENA	INTEGER8	—	—	Ja
362E	IN2PM	INTEGER8	—	—	Ja
362F	KTN	INTEGER32	⊠	—	Ja
3630	INPT	INTEGER16	—	—	Ja
3631	UCOMP	INTEGER32	—	—	Ja
3632	COLDSTART	—	—	Disabled	—
3633	reserviert				
3634	UID1	INTEGER32	—	—	Ja
3635	SETVCT	INTEGER16	—	—	Nein
3636	WPOS	INTEGER8	—	Disabled + Reset	Nein
3637	SRND	INTEGER32	—	—	Ja
3638	ERND	INTEGER32	—	—	Ja
3639	MDRV	INTEGER8	—	—	Ja
363A	BCC	INTEGER16	—	—	Nein
363B	FPGA	INTEGER8	—	Disabled + Reset	Ja
363C	REFMODE	INTEGER8	—	—	Ja
363D	VLO	INTEGER32	⊠	—	Ja
363E	WMASK	INTEGER32	—	—	Nein
363F	WPOSE	INTEGER32	—	—	Nein
3640	WPOSP	INTEGER32	—	—	Nein
3641	WPOSX	INTEGER32	—	—	Nein
3642	MOVE	INTEGER16	—	Enabled	—
3643	POSRSTAT	INTEGER32	—	—	Nein
3644	P1	INTEGER32	—	—	Ja
3645	P2	INTEGER32	—	—	Ja
3646	P3	INTEGER32	—	—	Ja
3647	P4	INTEGER32	—	—	Ja
3648	P5	INTEGER32	—	—	Ja
3649	P6	INTEGER32	—	—	Ja
364A	P7	INTEGER32	—	—	Ja
364B	P8	INTEGER32	—	—	Ja
364C	P9	INTEGER32	—	—	Ja
364D	P10	INTEGER32	—	—	Ja
364E	P11	INTEGER32	—	—	Ja

Nummer	ASCII-Kommando	Datenformat	Wichtung	Status	EEPROM
364F	P12	INTEGER32	—	—	Ja
3650	P13	INTEGER32	—	—	Ja
3651	P14	INTEGER32	—	—	Ja
3652	P15	INTEGER32	—	—	Ja
3653	P16	INTEGER32	—	—	Ja
3654	PTARGET	INTEGER32	—	—	Ja
3655	ACTRS232	INTEGER8	—	—	Nein
3656	ROFFS2	INTEGER32	—	—	Ja
3657	FW	INTEGER32	⊠	—	Nein
3658	reserviert				
3659	ACCUNIT	INTEGER32	—	—	Ja
365A	VCOMM	INTEGER32	—	—	Ja
365B	MTMUX	INTEGER16	—	—	Nein
365C	ROFFS0	INTEGER32	—	—	Ja
365D	REFLS	INTEGER32	—	—	Ja
365E	BOOT	—	—	—	Ja
365F	VUNIT	INTEGER32	—	—	Ja
3660	PUNIT	INTEGER32	—	—	Ja
3661	reserviert				
3662	reserviert				
3663	reserviert				
3664	reserviert				
3665	reserviert				
3666	reserviert				
3667	reserviert				
3668	reserviert				
3669	reserviert				
366A	reserviert				
366B	reserviert				
366C	reserviert				
366D	reserviert				
366E	TBRAKE	INTEGER16	—	—	Ja
366F	TBRAKE0	INTEGER16	—	—	Ja
3670	CMDDLY	INTEGER16	—	—	Ja
3671	reserviert				
3672	DRVCNFG	INTEGER32	—	—	Ja

VI Anhang

VI.1 Beispiele für die Inbetriebnahme

Alle Angaben sind hexadezimal. Die achsbezogenen Angaben beziehen sich immer auf die Station1.

VI.1.1 Wichtige Konfigurationsparameter für den CAN-Bus - Betrieb

Die folgenden Parameter sind für den CAN - Betrieb wichtig:

1. CBAUD (SDO 3515_h Subindex 01_h): Übertragungsrate für den CAN-Bus

2. ADDR (SDO 3505_h Subindex 01_h): Mit dem Kommando ADDR wird die Feldbus-Adresse des Verstärkers definiert. Nach der Änderung der Adresse sollten alle Parameter im EEPROM abgespeichert werden und der Verstärker aus- und eingeschaltet werden.

3. AENA (SDO 3506_h Subindex 01_h): Hiermit kann der Zustand des Software-Enable beim Einschalten des Verstärkers definiert werden. Mit dem Software-Enable wird einer externen Steuerung die Möglichkeit gegeben die Endstufe softwaremässig zu sperren bzw. freizugeben. Bei Geräten, die mit einem analogen Sollwert arbeiten (OPMODE=1,3) wird beim Einschalten des Verstärkers das Software-Enable automatisch gesetzt, so dass diese Geräte sofort betriebsbereit sind (Hardware-Enable vorausgesetzt). Bei allen anderen Geräten wird beim Einschalten das Software-Enable auf den Wert von AENA gesetzt. Die Variable AENA hat auch eine Funktion beim Resettieren des Verstärkers nach einem Fehler (über digitalen Eingang 1 bzw. mit dem ASCII-Kommando CLRFAULT). Bei Fehlern, die softwaremässig resettiert werden können, wird, nachdem der Fehler gelöscht wurde, der Software-Enable auf den Zustand von AENA gesetzt. Auf diese Weise ist das Verhalten des Verstärkers beim Software-Reset analog zu dem Einschaltverhalten.

4. DRVCNFG (SDO 3672_h Subindex 01_h): Mit der Konfigurationsvariable DRVCNFG können unterschiedliche CANopen-Zusatzfunktionen des Verstärkers aktiviert werden.

Bit0	=1	CANopen Einschalt-Telegramm 0 Bytes lang
	=0	CANopen Einschalt-Telegramm 8 Bytes lang
Bit1	=1	Beeinflussung der CANopen-Zustandsmaschine durch Enable/Disable. Die CANopen-Zustandsmaschine richtet sich nach dem internen Zustand des Servoverstärkers. Bei einer Änderung dieses Zustandes (z.B. Hardware-Disable) wird die CANopen-Zustandsmaschine automatisch aktualisiert (mit entsprechender Emergency-Meldung).
	=0	Die CANopen-Zustandsmaschine wird nicht beeinflusst.
Bit2	=1	Überprüfung der SDO-Länge, Generierung eines Emergency-Objektes bei fehlerhafter SDO-Länge
	=0	die SDO-Länge wird nicht überprüft.
Bit3	=1	Bei dieser Konfiguration können alle PDO-Mappings und Kommunikationsparameter abgespeichert werden. Dies erfolgt mit dem SDO 1010 sub 2 (-> Kap. IV.2.1.16). Bei Einschalten des Geräts wird dieses gespeicherte Mapping eingestellt.
	=0	Beim Einschalten des SERVOSTAR wird die Defaultkonfiguration der PDO - Mappings eingestellt.

5. MDRV (SDO 3639h, Subindex 01h):
Mit der Konfigurationsvariablen MDRV kann der Multidrive - Modus (MDRV = 1) für die Setup-Software eingestellt werden. Dann sind nur die ersten drei TPDOs und RPDOs nutzbar. Falls alle vier TPDOs und RPDOs genutzt werden sollen, muss diese Konfigurationsvaraibale auf 0 gesetzt sein.

VI.1.2 Prinzipielle Prüfung des Verbindungsaufbaus der Steuerung zumSERVOSTAR

Beim Einschalten des SERVOSTAR wird, abhängig von der Einstellung des Bit 2 des Parameters DRVCNFG, eine Emergency - Message mit 0 oder 8 Datenbytes (mit Inhalt 0) auf den Bus gesendet. Wenn sich im Bussystem kein geeigneter Empfänger findet, wird dieses Telegramm fortlaufend weiter gesendet.
Kann ein CAN - Master diese Nachricht nicht erkennen, können folgende Maßnahmen zur Überprüfung der Kommunikation durchgeführt werden:

- Überprüfung des Buskabels: richtiger Wellenwiderstand, korrekte Abschlusswiderstände an beiden Enden?
- Multimeterüberprüfung des Ruhepegels der Busleitungen CAN-H und CAN-L gegenüber der CAN-GND - Leitung (ca. 2.5 V).
- Oszilloskop - Überprüfung der Ausgangssignale an CAN-H und CAN-L am SERVOSTAR. Werden Signale auf den Bus gesendet? Spannungsdifferenz zwischen CAN-H und CAN-L bei logischer "0" ca. 2-3 V.
- Werden die Signale bei angeschlossenem Master nicht mehr weitergesendet? Eventuell Masterhardware überprüfen.
- Überprüfung der Master-Software!

VI.1.3 Beispiel für die Bedienung der Zustandsmaschine

Nach dem Einschalten des SERVOSTAR und dem Erkennen der Boot-Up-Message kann die Kommunikation über SDOs aufgenommen werden, z.B. können so Parameter abgefragt oder geschrieben werden oder die Zustandsmaschine des Antriebs gesteuert werden.

In den weiteren Beispielen wird angenommen, dass es Funktionen zum Lesen und Schreiben von SDOs gibt, die wie folgt aussehen:

SDO-Read (UINT Index, USHORT Subindex);

SDO-Write (UINT Index, USHORT Subindex, ULONG Wert);

Der Zustand der Zustandsmaschine kann über folgende Abfrage erkannt werden:

SDO-Read (6041_h, 00_h)

Direkt nach dem Einschalten erhält man dann z.B. als Antwortwert ein 0040_h. Dies entspricht dem Zustand „Switch on disabled“ (⇒ Kap. IV.4.1.1).

Auf dem CAN-Bus würde man als Daten folgendes sehen:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	40	41	60	00_h	00 00 00 00	
581	40	41	60	00_h	40 00 00 00	Antworttelegramm
	2 Byte Daten				Status	

Ist die Leistungsspannung vorhanden und das Hardware - Enable liegt auf High - Signal (24 V gegen DGND) kann durch SDO-Write (6040_h, 00_h, 0x7) versucht werden den Antrieb in den Zustand „Switched on“ zu schalten. Im Erfolgsfall wird dies in der SDO - Antwort positiv quittiert (Controlbyte 0 im Datenfeld = 60_h).

Switch on

Die Nachrichten sehen dann wie folgt aus:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	40	60	00_h	07 00 00 00	controlword
581	60	40	60	00_h	00 00 00 00	Antworttelegramm

controlword = 0x0007 Bedeutung: Bit 0, Bit 1, Bit 2 gesetzt ⇒ Switch On, Disable Voltage off, Quick Stop off

Statusabfrage 2

Der neue Zustand kann dann wieder abgefragt werden und liefert das folgende Ergebnis:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	40	41	60	00_h	—	Status abfragen
581	4B	41	60	00_h	23 00 00 00	Antworttelegramm

Status = 0x0023 Bedeutung: Bit 0, Bit 1, Bit 5 gesetzt ⇒ ready to Switch On, Switched On, Quick Stop

VI.1.4 Beispiel für PDO-Bedienung

Es sollen alle vier möglichen PDOs im Betrieb verwendet werden:

1. RPDO: PDO Trajektorie für eine Achse
2. RPDO: PDO Steuerwort und Modeumschaltung
1. TPDO: PDO erweiterter Status
2. TPDO: PDO mit Inkrementeller Istposition, Drehzahl und Betriebsartanzeige

Vorgehensweise:

Da das erste RPDO nicht in der geforderten Form vom Antrieb vordefiniert zur Verfügung gestellt wird (⇒ Kap. IV.3.1) muß es selbst zusammengestellt werden. Dazu ist vorweg zu prüfen, ob die Einträge für die inkrementelle Sollwertvorgabe mappbar zur Verfügung steht.
Das ist bei SDO 2022_h Subindex 04_h der Fall. Also wird das 1. RPDO selektiert:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	26	00_h	25 00 00 00	PDO 37 als 1. RPDO auswählen
581	60	00	26	00_h	00 00 00 00	Antworttelegramm

Damit ist das frei mappbare RPDO 37 ausgewählt. Im nächsten Schritt muß dieses PDO mit Daten versehen werden. Dies geschieht über die Mappingparameter für das erste RPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	00	16	00_h	00 00 00 00	Daten für 1. RPDO löschen
581	60	00	16	00_h	00 00 00 00	Antworttelegramm
601	23	00	16	01_h	20 04 22 20	Daten für ersten Eintrag des 1. RPDO SDO 2022_h Subindex 04_h, Datenlänge 32 Bit
581	60	00	16	01_h	00 00 00 00	Antworttelegramm

Damit ist dieses erste PDO datenmäßig festgelegt, es enthält vier Byte Nutzdaten.

Jetzt können die Kommunikationsparameter festgelegt werden:

Es soll standardmäßig auf COB-ID 201_h reagieren. Daher muß der Subindex 01_h auf seinem Defaultwert bleiben. Aber der Antrieb soll auf jedes SYNC-Objekt reagieren, daher muß bei Subindex 2 ein Wert von 1 vorgegeben werden. Also:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	14	02_h	01 00 00 00	Reaktion auf jedes SYNC
581	60	00	14	02_h	00 00 00 00	Antworttelegramm

Das zweite RPDO soll zwei Komponenten enthalten, das CANopen Controlwort (SDO 6040_h Subindex 00_h) und das SDO zur Änderung der Betriebsart (SDO 6060_h Subindex 00_h).

Die Auswahl des 2. RPDO sieht dann wie folgt aus:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	26	00_h	26 00 00 00	PDO 38 als 2. RPDO auswählen
581	60	01	26	00_h	00 00 00 00	Antworttelegramm

Daraufhin wird das Mapping festgelegt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	16	00_h	00 00 00 00	Daten für 2. RPDO löschen
581	60	01	16	00_h	00 00 00 00	Antworttelegramm
601	23	01	16	01_h	10 00 40 60	Daten für ersten Eintrag des 2. RPDO SDO 6040_h Subindex 00_h, Datenlänge 16 Bit
581	60	01	16	01_h	00 00 00 00	Antworttelegramm
601	23	01	16	02_h	08 00 60 60	Daten für zweiten Eintrag des 2. RPDO SDO 6060_h Subindex 00_h, Datenlänge 8 Bit
581	60	01	16	02_h	00 00 00 00	Antworttelegramm

Dieses Objekt soll sofort ausgewertet werden, daher können die Kommunikationsparameter auf ihren Defaultwerten bleiben.

Das erste TPDO ist im Antrieb schon vorhanden, es muß nur ausgewählt werden.

Es wird das PDO 23 ausgewählt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	2A	00_h	17 00 00 00	PDO 23 als 1. TPDO auswählen
581	60	00	2A	00_h	00 00 00 00	Antworttelegramm

Das Mapping dazu kann über das SDO $1A00_h$ ausgelesen werden. Das PDO enthält 2 Byte für das CANopen-Zustandswort und 4 Byte für das Herstellerstatusregister.

Das zweite TPDO wird wieder zusammengestellt:

1. Die Anwahl über SDO 2A01:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	2A	00_h	25 00 00 00	PDO 37 als 1. TPDO auswählen
581	60	01	2A	00_h	00 00 00 00	Antworttelegramm

2. Das Mapping der drei geforderten Komponenten:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	1A	00_h	00 00 00 00	Daten für 2. TPDO löschen
581	60	01	1A	00_h	00 00 00 00	Antworttelegramm
601	23	01	1A	01_h	20 03 70 20	Daten für ersten Eintrag des 2. TPDO SDO 2070_h Subindex 03_h, Datenlänge 32 Bit
581	60	01	1A	01_h	00 00 00 00	Antworttelegramm
601	23	01	1A	02_h	18 02 70 20	Daten für zweiten Eintrag des 2. TPDO SDO 2070_h Subindex 02_h, Datenlänge 24 Bit
581	60	01	1A	02_h	00 00 00 00	Antworttelegramm
601	23	01	1A	02_h	08 00 61 60	Daten für dritten Eintrag des 2. TPDO SDO 6060_h Subindex 00_h, Datenlänge 8 Bit
581	60	01	1A	02_h	00 00 00 00	Antworttelegramm

Jetzt können die Kommunikationsparameter festgelegt werden:

Der Antrieb soll auf jedes SYNC-Objekt reagieren, daher muß bei Subindex 2 ein Wert von 1 vorgegeben werden. Also:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	18	02_h	01 00 00 00	2. TPDO: Reaktion auf jedes SYNC
581	60	01	18	02_h	00 00 00 00	Antworttelegramm

VI.1.5 Beispiel für die Referenzierung

Beim Betrieb des SERVOSTAR als Linearachse muß, bevor Positionierungen durchgeführt werden können, ein Referenzpunkt festgelegt werden. Dies kann über ein einfaches "Referenzpunkt setzen" (Bit 12 Steuerwort = 0 -> 1 -> 0 bzw. Referenzfahrt 35 im Mode Homing) oder über das Starten einer Referenzfahrt entweder im herstellerspezifischen Mode "Referenzierung" (0xF9) oder im Homing Mode (0x6) erfolgen.

Hier wird beispielhaft das Vorgehen im Mode "Referenzierung" aufgezeigt.

Dazu wird zuerst in den Modus "Referenzfahrt" geschaltet:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	60	60	00_h	F9 00 00 00	Mode Referenzierung
581	60	60	60	00_h	00 00 00 00	O.K. - Meldung

Im folgenden werden alle die Referenzfahrt betreffenden Parameter über den Bus eingestellt. Wenn man sicher sein kann, dass niemand die Parameter im Gerät geändert hat, kann dieser Teil entfallen, da die Geräte die Daten nullspannungssicher speichern können. (Vorher müssen die Eingänge als Endschalter konfiguriert sein.)

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	2A	01_h	17 00 00 00	Referenzfahrt auf Endschalter & Resolvernullpunkt
581	60	24	20	01_h	00 00 00 00	OK
601	2F	24	20	02_h	00 00 00 00	Negative Fahrtrichtung
581	60	24	20	02_h	00 00 00 00	OK
601	2B	24	20	03_h	10 27 00 00	v = 10mm/s
581	60	24	20	03_h	00 00 00 00	OK
601	2B	24	20	04_h	32 00 00 00	Anfahrrampe 50ms
581	60	24	20	04_h	00 00 00 00	OK
601	2B	24	20	05_h	32 00 00 00	Bremsrampe 50ms
581	60	24	20	05_h	00 00 00 00	OK
601	23	24	20	06_h	30 75 00 00	Referenzoffset 30000µm
581	60	24	20	06_h	00 00 00 00	OK

Zur Überprüfung der Bitsignale, die für den Referenzfahrtablauf wichtig sind, soll hier das PDO erweiterter Status (PDO 23) verwendet werden.

Dazu wird zuerst das PDO 23 als TPDO 1 ausgewählt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	2A	00_h	17 00 00 00	PDO 23 als 1. TPDO auswählen
601	60	00	2A	00_h	00 00 00 00	Antworttelegramm

Das TPDO1 setzt sich jetzt also aus 6 Bytes zusammen, wobei die ersten zwei Bytes das CANopen Zustandswort (SDO 6041) enthalten, die anderen vier Bytes das herstellerspezifische Statusregister (SDO 1002).

Diese Belegung kann über die SDOs für das PDO-Mapping (SDO 1A00$_h$, Subindex 00$_h$-02$_h$) abgefragt werden. Danach werden die PDOs über ein NMT - Objekt freigegeben:

COB-ID	Command specifier (CS)	Node-ID
0	1	1

Ab diesem Zeitpunkt werden Zustandsänderungen am TPDO1 automatisch gemeldet.

Nun kann die Referenzfahrt mit Bit 4 des CANopen - Controlwords gestartet werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2B	40	60	00$_h$	1F 00 00 00	Mode
581	60	40	60	00$_h$	00 00 00 00	Referenzfahrt läuft bis Referenzbedingung erfüllt

Mit der steigenden Flanke des Bits "Referenzpunkt gesetzt" im herstellerspezifischen Statusregisters kann erkannt werden, dass der Sevoantrieb sein Positionssystem geeicht hat. Der Vorgang der Referenzierung ist mit Rücksetzen des Bits "Fahrauftrag aktiv" beendet.

Ein TPDO1 könnte also wie folgt aussehen:

COB-ID	Daten	Kommentar
181	05 27 00 00 0A 54	Mode

Der Zustand der Referenzfahrt lässt sich über das erweiterte Statusregister, Bit 17 (Referenzpunkt gesetzt) erkennen.

VI.1.6 Beispiel für Fahrsatzverarbeitung

Lageregelung einschalten

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	60	60	00$_h$	FF 00 00 00	Lageregelmode
581	60	60	60	00$_h$	00 00 00 00	Lageregelung eingeschaltet

Zweites Receive-PDO mappen

(Starte Fahrsatz, Fahrsätze für Beispiel schon definiert. 1. Receive-PDO steht standardmäßig auf dem controlword.)

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	26	00$_h$	23 00 00 00	Starte Fahrsatz Objekt
581	60	01	26	00$_h$	00 00 00 00	OK, gemappt

NMT-Zustandsmaschine auf “operational” schalten

COB-ID	Command specifier (CS)	Node-ID
0	1	1

Zweites Receive-Objekt ansprechen

COB-ID	Fahrsatznummer-Low	Fahrsatznummer-High
301	01	00

Antwort: keine, angegebener Fahrsatz 1 wird abgearbeitet

Motor Quick Stop

COB-ID	Control-Low	Control-High
201	03	00

Antwort: keine, Motor wird mit t_not angehalten

Regler disablen

COB-ID	Control-Low	Control-High	Fahrsatznummer
201	03	00	1

Antwort: keine, Antrieb wird drehmomentfrei

VI.1.7 Beispiel für Bedienung des Profile-Position Modes

Dieses Kapitel zeigt die Bedienung des Profile-Position Modes. Dazu werden die PDOs wie folgt voreingestellt:

Erstes RPDO: PDO Steuerwort (Nr. 1)
Zweites RPDO: frei mappbares PDO 2 (Nr. 38)
Erstes TPDO: frei mappbares PDO 1 (Nr. 37)
Zweites TPDO: frei mappbares PDO 2 (Nr. 38)

Die Telegramme sehen analog zum Beispiel für PDO-Bedienung (⇒ Kap. VI.1.4) aus.

In die frei mappbaren PDOs werden Daten entsprechend der folgenden Beispieltelegramme gelegt:

1.: zweites RPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	16	00_h	00 00 00 00	2. RPDO: Mapping löschen
581	60	01	16	00_h	00 00 00 00	
601	23	01	16	01_h	20 00 7A 60	2. RPDO Eintrag 1: tar-get_position
581	60	01	16	01_h	00 00 00 00	
601	23	01	16	02_h	20 00 81 60	2. RPDO Eintrag 2: profi-le_velocity
581	60	01	16	02_h	00 00 00 00	

2.: erstes TPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	00	1A	00_h	00 00 00 00	1. TPDO: Mapping löschen
581	60	00	1A	00_h	00 00 00 00	
601	23	00	1A	01_h	10 00 41 60	1. TPDO Eintrag 1: Profil-zustandswort
581	60	00	1A	01_h	00 00 00 00	
601	23	00	1A	02_h	08 09 80 20	1. TPDO Eintrag 2: TRJSTAT 3. Byte = Herst.Status 3. Byte
581	60	00	1A	02_h	00 00 00 00	

3.: zweites TPDO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	1A	00_h	00 00 00 00	2. TPDO: Mapping löschen
581	60	01	1A	00_h	00 00 00 00	
601	23	01	1A	01_h	20 00 64 60	2. TPDO Eintrag 1: positi-on_actual_value
581	60	01	1A	01_h	00 00 00 00	
601	23	01	1A	02_h	20 00 6C 60	1. TPDO Eintrag 2: veloci-ty_actual_value
581	60	01	1A	02_h	00 00 00 00	

Das erste TPDO soll ereignisgesteuert gesendet werden. Da dies dem Defaultwert der Kommunikationsparameter entspricht, braucht hier nichts verändert zu werden. Das zweite TPDO soll mit jedem SYNC vom Antrieb gesendet werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	01	18	02_h	01 00 00 00	2. TPDO mit jedem SYNC versenden
581	60	01	18	02_h	00 00 00 00	

Nach der Festlegung der PDOs können diese mit dem NMT freigegeben werden:

COB-ID	Command specifier (CS)	Node-ID
0	1	1

Falls die mechanische Auflösung vorgegeben werden soll, kann diese nun über SDO 6093_h, Subindex 01_h und 02_h geschrieben werden. Die Voreinstellung nach dem Einschalten des Antriebs entspricht den antriebspezifischen Faktoren PGEARI und PGEARO:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	93	60	01_h	00 00 10 00	2^{20} Inkremente
581	60	93	60	01_h	00 00 00 00	
601	23	93	60	02_h	A0 8C 00 00	3600 Nutzereinheiten
581	60	93	60	02_h	00 00 00 00	

Im Beispiel könnte man zum Beispiel einen Rundtisch mit einer Winkelauflösung von 0,01 Grad bedienen.

Nach diesen Einstellungen kann eine Referenzfahrt eingestellt und angestoßen werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2B	40	60	00_h	0F 00 00 00	Controlword: Operation Enable
581	60	40	60	00_h	00 00 00 00	
601	2F	60	60	00_h	06 00 00 00	Betriebsart: Homing Mode einstellen
581	60	60	60	00_h	00 00 00 00	
601	2F	98	60	00_h	0C 00 00 00	Referenzfahrt 12, negative Fahrtrichtung
581	60	98	60	00_h	00 00 00 00	
601	23	99	60	01_h	40 19 01 00	Referenzfahrtgeschw. 72000 Einh./s = 2 $s^{-!}$
581	60	99	60	01_h	00 00 00 00	
601	2B	40	60	00_h	1F 00 00 00	Referenzfahrt starten
581	60	40	60	00_h	00 00 00 00	

Nach dem Start der Referenzfahrt könnten dann folgende Telegramme des 1. TPDO kommen:

COB-ID	Byte 0	Byte 1	Byte 2
181	27	00	21
181	27	10	27
181	27	10	23
181	27	14	2B
181	27	15	2A

Für die Erkennung des Endes der Referenzfahrt sind folgende Bits des Bytes 2 erforderlich: Bit 0 = 0 Referenzfahrt abgeschlossen, Bit 1 = 1 Referenzpunkt gesetzt, Bit 3 = 1 In Position. Anschließend wird die Referenzfahrt auch über das Steuerwort beendet, diesmal über das 1. RPDO:

COB-ID	Byte 0	Byte 1
201	0F	00

Nun kann in den Profile Position Mode geschaltet werden und Rampen für Positionierungen vorgegeben werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	23	83	60	00_h	32 00 00 00	50ms Beschleunigungszeit
581	60	83	60	00_h	00 00 00 00	
601	23	84	60	00_h	32 00 00 00	50ms Bremszeit
581	60	84	60	00_h	00 00 00 00	
601	2F	60	60	00_h	01 00 00 00	Profile Position Mode
581	60	60	60	00_h	00 00 00 00	

Eine Positionierung kann nun über Vorgabe der Sollwerte über das 1. RPDO und anschließenden Start über das 2. RPDO. Dabei kommt der Handshake über New Setpoint (Controlword) und Setpoint acknowledge (status word) zum Einsatz.

a.: Sollwert

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
301	F4	01	00	00	E8	03	00	00

b.: Controlword mit „new setpoint“ – Bit (Bit 4) gesetzt

COB-ID	Byte 0	Byte 1
201	1F	00

c.: Warten, dass das CANopen status word „setpoint acknowledge“ (Bit 12) meldet: (z.B.)

COB-ID	Byte 0	Byte 1	Byte 2
181	27	15	03

d.: Controlword mit „new setpoint“ – Bit (Bit 4) zurückgesetzt

COB-ID	Byte 0	Byte 1
201	0F	00

e.: Antrieb nimmt setpoint acknowldege zurück

COB-ID	Byte 0	Byte 1	Byte 2
181	27	01	03

Auf Abschluß der Positionierung warten

COB-ID	Byte 0	Byte 1	Byte 2
181	27	05	0A

VI.1.8 ASCII - Kommunikation

Die ASCII-Kommunikation findet Sinnvollerweise über PDOs statt, da sie so effizienter genutzt werden kann. Dazu muss sich die NMT-Zustandmaschine im Zustand “operational” befinden.

Beispiel: Parameter T-Tacho (siehe Handbuch Inbetriebnahme-Software DRIVE.EXE) lesen. (Alle Daten Hexadezimal mit ASCII-Entsprechung in eckigen Klammern darunter)

Richtung	COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Master ⇒ SERVOSTAR	301	47_h [G]	56_h [V]	46_h [F]	42_h [B]	54_h [T]	$0D_h$ [CR]	$0A_h$ [LF]	0_h [NUL]
SERVOSTAR ⇒ Master	281	30_h [0]	$2E_h$ [.]	36_h [6]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]
SERVOSTAR ⇒ Master	281	$0D_h$ [CR]	$0A_h$ [LF]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]
SERVOSTAR ⇒ Master	281	$2D_h$ [-]	$2D_h$ [-]	$3E_h$ [>]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]	0_h [NUL]

Erklärung: In Telegramm 1 fragt der Master den Parameter “GVFBT” an, abgeschlossen von den ASCII-Codes “CR LF”. Das freie letzte Byte wird mit ”NUL” gefüllt.
Die Antwort des SERVOSTAR erfolgt in Telegramm 2 mit dem Wert “0.6”, dem Abschlusscode “CR LF” und dem Prompt für den nächsten Parameter bzw. das nächste Kommando “—>”. Die Segmentierung der Antwort in drei Telegramme erfolgt nicht zwingend, sondern hängt von der eingestellten Übertragungsrate und internen Synchronisationsmechanismen ab.

VI.1.9 Test für Synctelegramme

Aufgabenstellung:

1. PDO mit Starte Fahrsatz belegen (1st Receive PDO)
2. PDO mit Ist - Lage (PDO21) belegen (1st Transmit PDO), mit jedem 2. SYNC auslösen
3. PDO mit statusword (PDO1) belegen (2nd Transmit PDO), mit jedem 3. SYNC auslösen

Telegramme mit jeweiligen Antworten:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	00	26	00_h	23 00 00 00	PDO Starte Fahrsatz auf 1. RPDO setzen
581	60	00	26	00_h	00 00 00 00	
601	2F	00	2A	00_h	16 00 00 00	PDO Ist-Lage auf 1. TPDO setzen
581	60	00	2A	00_h	00 00 00 00	
601	2F	01	2A	00_h	17 00 00 00	PDO erweitertes status-word auf 2. TPDO setzen
581	60	01	2A	00_h	00 00 00 00	
601	2F	00	18	02_h	02 00 00 00	1. TPDO auf Trigger, durch jedes 2. SYNC ersetzen
581	60	00	18	02_h	00 00 00 00	
601	2F	01	18	02_h	03 00 00 00	2. TPDO auf Trigger, durch jedes 3. SYNC ersetzen
581	60	01	18	02_h	00 00 00 00	

VI.1.10 SYNC-Objekt

COB-ID
080

Bedeutung: Bei jedem zweiten SYNC kommt das Objekt 181 (TPDO 1), bei jedem dritten SYNC kommt das Objekt 281 (TPDO 2).

VI.1.11 Emergency-Objekt

Zieht man zwischendurch zum Beispiel den Resolverstecker ab, löst man einen schweren Fehler im Regler aus. Dies führt zu einem Emergency - Telegramm.

COB-ID	Emergency error code		Error register		
	Low	High			
081	10	43	08	00 00 00 00	Motortemperatur, temperature, manu-
081	00	00	88	00 00 00 00	facturer specific

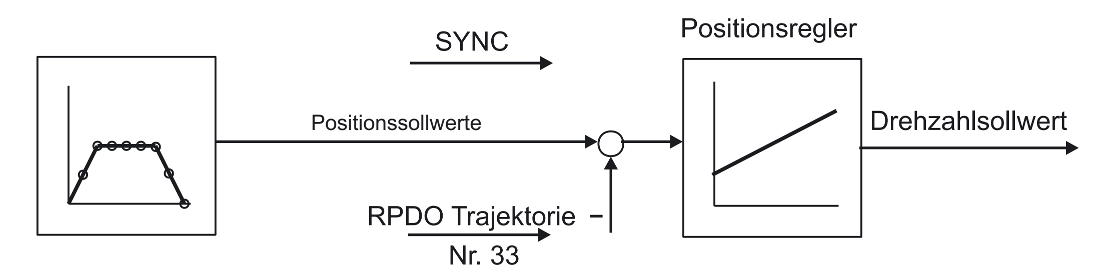
VI.2 Spezielle Applikationen

VI.2.1 Externe Trajektorie

VI.2.1.1 Lageregler im Servoverstärker

In diesem Beispiel wird der Fall betrachtet, in dem 2 Achsen jeweils Positionssollwerte über das RPDO 33 Trajektorie erhalten.

Reglerstruktur bei Lageregler im Servoverstärker:



Position + Status
TPDO Inkrementeller Positionsistwert
Nr. 33

VI.2.1.1.1 Beschreibung

Alle Angaben sind hexadezimal. Die beiden Achsen im System haben im Beispiel die Stationsadressen 1 und 2.

Beispieltelegramme und -antworten:

Voraussetzungen:

- Es muss für beide Achsen definiert werden welche Auflösungen innerhalb einer Motorumdrehung verwendet werden sollen: PRBASE auf 16 oder 20 Bit pro Umdrehung.
- Das Zeitraster für Trajektorienvorgabe muss über den Parameter PTBASE eingestellt werden. Dabei entspricht die Einheit einem Wert von 250 Mikrosekunden, d.h. z.B. PTBASE = 8 ergibt ein Trajektorienzeitwert von 2 Millisekunden.
- Die Parameter müssen im EEPROM gespeichert werden.
- Die gespeicherten Werte stehen erst nach einem Neustart zur Verfügung.

Das PDO Trajektorie enthält 2 Trajektoriensollwerte und kann an mehrere Stationen gleichzeitig gesendet werden, wobei sich jede Station ihre Trajektoriendaten herausholen kann.

Zweites Receive-PDO für beide Achsen auf das RPDO 33 Trajektorie (33_d = 21_h) mappen:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	26	00_h	21 00 00 00	PDO Trajektorie für 1. Achse eingestellt
581	60	01	26	00_h	00 00 00 00	
602	2F	01	26	00_h	21 00 00 00	PDO Trajektorie für 2. Achse eingestellt
582	60	01	26	00_h	00 00 00 00	

Nachdem das PDO Trajektorie nun auf zwei Achsen gemappt ist, müssen die Kommunikationsparameter bei beiden so eingestellt werden, dass sie auf denselben Kommunikationsobjekt-Identifier (COB-ID) reagieren. Der COB-ID für die erste Station kann dabei auf seinem Defaultwert 301 bleiben, der für die zweite Station kann dann auf diesen umgemappt werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
602	23	01	14	01_h	01 03 00 00	2. RPDO für 2. Achse auf 301 mappen
582	60	01	14	01_h	00 00 00 00	

Beide Stationen reagieren jetzt auf denselben COB-ID 301.

Mit dem SDO 2721_h Subindex 00_h kann dann bestimmt werden aus welchem Teil des 8 Byte-Datenfelds jede Achse ihre Trajektorie bezieht. Mit dem Wert 0 werden dabei die Bytes 0..3 der Daten ausgewählt, mit dem Wert 1 die Bytes 4..7:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	21	27	00_h	00 00 00 00	1. Achse verwendet Byte 0..3 als Wert
581	60	21	27	00_h	00 00 00 00	
602	2F	21	27	00_h	01 00 00 00	2. Achse verwendet Byte 4..7 als Wert
582	60	21	27	00_h	00 00 00 00	

Die Istpositionen der Achsen sollen auch als inkrementelle Istpositionen an die Steuerung zurückkommen. Daher werden die jeweils zweiten Transmit-PDOs auf das TPDO 33 Inkrementeller Positionsistwert ($33_d = 21_h$) gemappt.:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	2A	00_h	21 00 00 00	PDO Trajektorie für 1. Achse eingestellt
581	60	01	2A	00_h	00 00 00 00	
602	2F	01	2A	00_h	21 00 00 00	PDO Trajektorie für 2. Achse eingestellt
582	60	01	2A	00_h	00 00 00 00	

Es wird hier davon ausgegangen, dass die beiden Geräte mit jedem SYNC-Kommando neue Trajektorienwerte übernehmen und ihre inkrementellen Positionsistwerte zurücksenden sollen. Also müssen die Kommunikationsparameter entsprechend eingestellt werden:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	01	14	02_h	01 00 00 00	RPDO 2 Achse 1 Reaktion auf jedes SYNC
581	60	01	14	02_h	00 00 00 00	
602	2F	01	14	02_h	01 00 00 00	RPDO 2 Achse 2 Reaktion auf jedes SYNC
582	60	01	14	02_h	00 00 00 00	
601	2F	01	18	02_h	01 00 00 00	TPDO 2 Achse 1 Reaktion auf jedes SYNC
581	60	01	18	02_h	00 00 00 00	
602	2F	01	18	02_h	01 00 00 00	TPDO 2 Achse 2 Reaktion auf jedes SYNC
582	60	01	18	02_h	00 00 00 00	

Damit Trajektorien gefahren werden können, müssen sich beide Servoverstärker im entsprechenden Modus befinden. Dies wird über den Index 6060_h eingestellt:

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2F	60	60	00_h	FA 00 00 00	Trajektorienmodus für Achse 1 einstellen
581	60	60	60	00_h	00 00 00 00	
602	2F	60	60	00_h	FA 00 00 00	Trajektorienmodus für Achse 2 einstellen
582	60	60	60	00_h	00 00 00 00	

Um die Achsen zu starten, müssen nun die Servoverstärker in den fahrbereiten Zustand (operational enable) gebracht und die Netzwerkmanagementfunktionen gestartet werden.

Die Netzwerkmanagementfunktionen, die die Verwendung der Prozessdatenobjekte freischalten, werden durch das folgende Telegramm für beide Achsen gestartet:

NMT (**N**etwork **M**anagemen**t**) - Zustandsmaschine auf „operational“ schalten

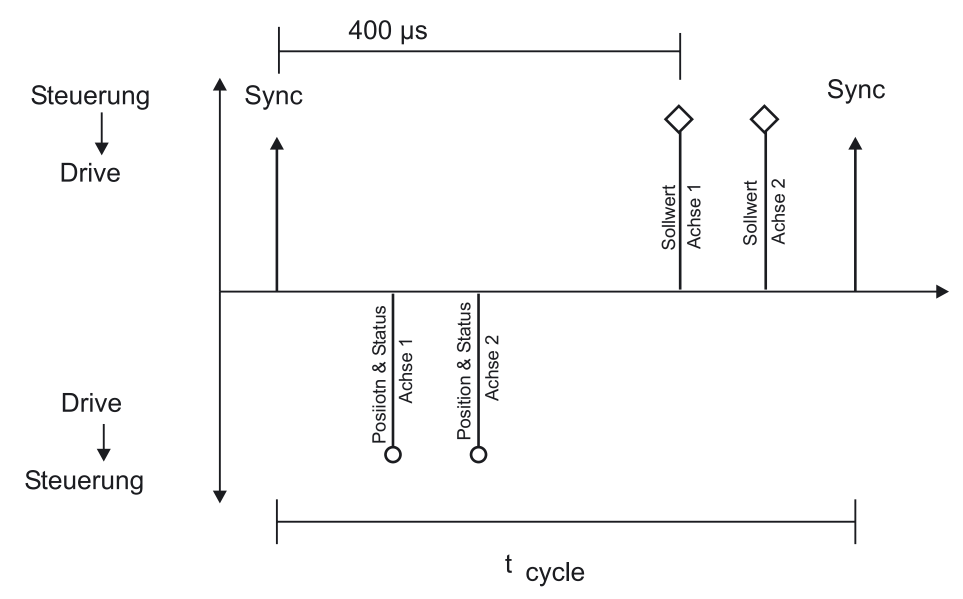
COB-ID	Command specifier (CS)	Node-ID	Kommentar
0	1	1	Alle Achsen NMT freigeben

Danach werden die Servoverstärker einzeln bestromt und in den Zustand „operation enable“ gebracht.

Steuerwort für Enable Operation

COB-ID	Control-Byte	Index		Subindex	Daten	Kommentar
		Low-Byte	High-Byte			
601	2B	40	60	00_h	0F 00 00 00	Steuerwort für Achse 1
581	60	40	60	00_h	00 00 00 00	
602	2B	40	60	00_h	0F 00 00 00	Steuerwort für Achse 2
582	60	40	60	00_h	00 00 00 00	

Durch die obige Konfiguration wird jetzt wie im folgenden Bild dargestrellt ein zyklischer Ablauf ermöglicht:



z.B. 2 Achsen

t_{cycle} 1 ms pro Achse bei 1 MBaud

Nun können über das RPDO 2 Trajektoriendaten für beide Achsen vorgegeben werden, z.B. wie folgt:

COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
301	F4	01	00	00	E8	03	00	00

In diesem Beispiel bekommt die erste Achse einen Trajektorienwert von 500 Inkrementen (Bytes 0..3) und die zweite Achse einen Trajektorienwert von 1000 Inkrementen.

Die Werte werden von den Achsen übernommen und mit Erhalt des nächsten SYNC – Telegramms positioniert.

Das SYNC – Telegramm sieht dabei wie folgt aus:

COB-ID
080

Beide Achsen senden anschließend ihre inkrementellen Positionen und ihre Statusregister beim Erhalt des SYNC-Objekts mit der COB-ID des 2. TPDOs zurück:

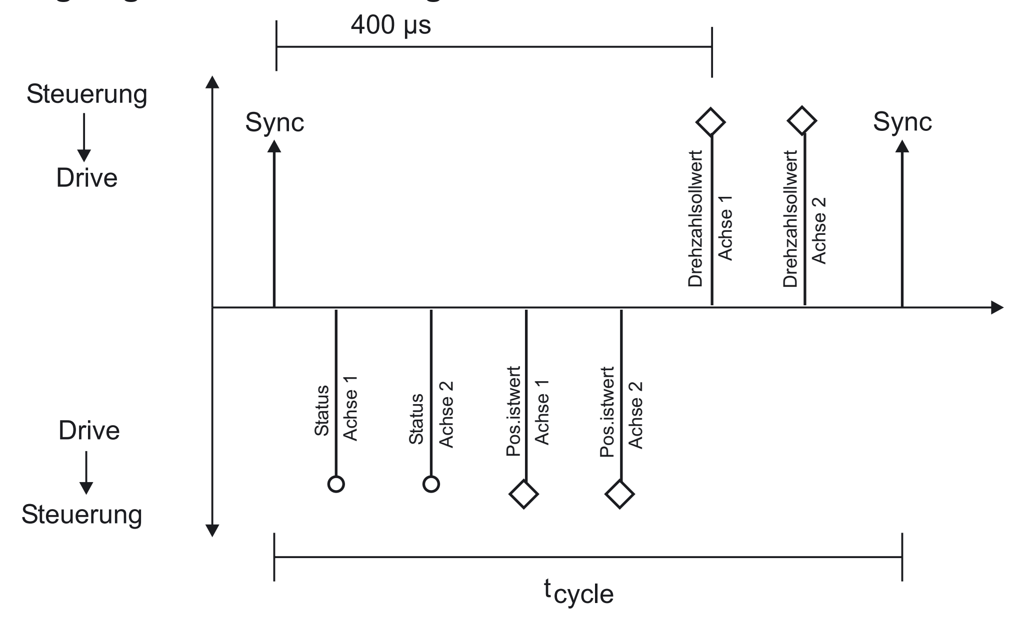
COB-ID	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Kommentar
181	23	01	00	00	00	00	03	44	Position und Hersteller-statusregister Achse1
182	A5	02	00	00	00	00	03	44	Position und Hersteller-statusregister Achse2

Tritt im Verlauf der Bedienung ein Fehler auf, wird von der betroffenen Achse eine Emergency-Message gesendet, die z.B. so aussehen kann:

Emergency – Objekt

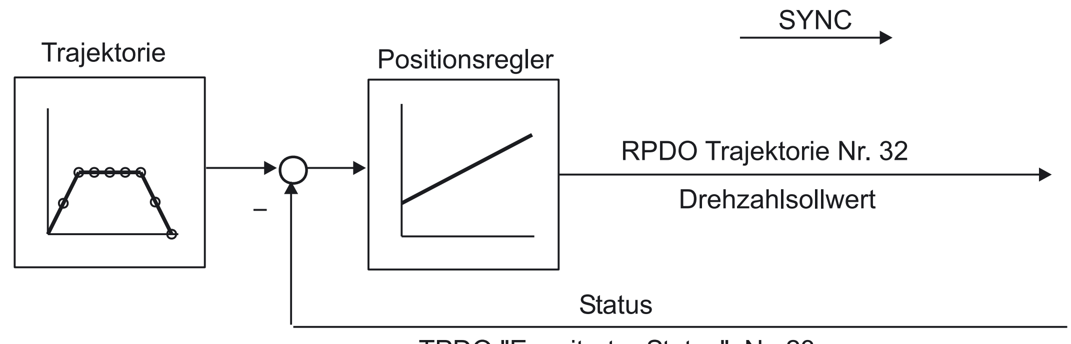
COB-ID	Emergency error code		Error register	Kategorie		
	Low	High				
081	10	43	08	01	00 00 00 00	Motortemperatur, temperature, manufacturer specific
081	00	00	08	00	00 00 00 00	

VI.2.1.2 Lageregler in der Steuerung



z.B. 2 Achsen

t_{cycle} 1 ms pro Achse bei 1 MBaud



VI.3 Beschreibung Objektverzeichnis

Die folgende Tabelle beschreibt das Objektverzeichnis (Object Dictionary).
Die Spalte "Def" weist auf die entsprechende Norm bzw. das herstellerspezifische Profil hin:
S = SERVOSTAR **3** = DS301 **4** = DS402

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
1000_h	—	3	UINTEGER32	r	—	Gerätetyp	—
1001_h	—	3	UINTEGER8	r	—	Fehlerregister	—
1002_h	—	3	UINTEGER32	r	—	Herstellerspezifisches Statusregister	—
1003_h	00_h	3	UINTEGER8	r	—	Pre-defined errror field (Anzahl der Einträge)	—
	01_h	3	UINTEGER32	r/w	—	Zuletzt gemeldeter Fehler	—
1004_h	00_h	3	UNSIGNED32	r	—	Zahl der unterstützen PDOs	—
	01_h	3	UNSIGNED32	r	—	Zahl der synchronen PDOs	—
	02_h	3	UNSIGNED32	r	—	Zahl der asynchronen PDOs	—
1005_h	—	3	UINTEGER32	r	—	COB-ID SYNC message	—
1006_h	—	3	UNSIGNED32	r/w	—	Zykluszeit der Kommunikation	—
1007_h	—	3	UNSIGNED32	r/w	—	Zeitfenster für synchrone CAN-Nachrichten	—
1008_h	—	3	Visible String	r	—	Gerätename	—
$100A_h$	—	3	Visible String	r	—	Softwareversion	VER*
$100B_h$	—	3	UINTEGER32	r	—	Knotenadresse	ADDR
$100C_h$	—	3	UINTEGER16	r/w	—	Guard time	—
$100D_h$	—	3	UINTEGER8	r/w	—	Life time factor	—
$100E_h$	—	3	UNSIGNED32	r/w	—	Node Guarding COB - Identifier	—
$100F_h$	—	3	UNSIGNED32	r/w	—	Zahl der unterstützten SDOs	—
1010_h	00_h	3	2	r	—	Store parameters / Anzahl der Einträge	—
	01_h	3	UNSIGNED32	r/w	—	Store parameters / save all parameters	—
	02_h	3	UNSIGNED32	r/w	—	Store parameters / save communication parameters	—
1012_h	—	3	UNSIGNED32	r/w	—	COB - ID der time stamp message (i.V.)	—
1013_h	—	3	UNSIGNED32	r/w	—	High resolution time stamp (i.V.)	—
1014_h	—	3	UNSIGNED32	r/w	—	COB - ID der Emergency message	—
1018_h	00_h	3	RECORD	r	—	Identity Object	—
	01_h	3	UINTEGER32	r	—	Vendor ID	—
	02_h	3	UINTEGER32	r	—	Product Code	—
	03_h	3	UINTEGER32	r	—	Revision number	—
	04_h	3	UINTEGER32	r	—	Serial number	SERIALNO
1400_h	00_h	3	RECORD	r	—	Empfangs- PDO 1 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1401_h	00_h	3	RECORD	r	—	Empfangs- PDO 2 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1402_h	00_h	3	RECORD	r	—	Empfangs- PDO 3 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1403_h	00_h	3	RECORD	r	—	Empfangs- PDO 4 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time (bei RPDOs nicht sinnvoll)	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1600_h	00_h	3	RECORD	r/w	—	Empfangs- PDO 1 Mappingparameter	—
	01_h-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
1601_h	00_h	3	RECORD	r/w	—	Empfangs- PDO 2 Mappingparameter	—
	01_h-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
1602_h	00_h	3	RECORD	r/w	—	Empfangs- PDO 3 Mappingparameter	—
	01_h-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
1603_h	00_h	3	RECORD	r/w	—	Empfangs- PDO 4 Mappingparameter	—
	01_h-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
1800_h	00_h	3	RECORD	r	—	Sende- PDO 1 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1801_h	00_h	3	RECORD	r	—	Sende- PDO 2 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1802_h	00_h	3	RECORD	r	—	Sende- PDO 3 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
1803_h	00_h	3	RECORD	r	—	Sende- PDO 4 Kommunikationsparameter	—
	01_h	3	UINTEGER32	r/w	—	PDO COB - ID	—
	02_h	3	UINTEGER8	r/w	—	Übertragungstyp	—
	03_h	3	UINTEGER16	r/w	—	Inhibit time	—
	04_h	3	UINTEGER8	r/w	—	Kompatibilitätseintrag (CMS priority group)	—
$1A00_h$	00_h	3	RECORD	r/w	—	Sende- PDO 1 Mappingparameter	—
	01_h-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
$1A01_h$	00_h	3	RECORD	r/w	—	Sende- PDO 2 Mappingparameter	—
	01-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
$1A02_h$	00_h	3	RECORD	r/w	—	Sende- PDO 3 Mappingparameter	—
	01-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
$1A03_h$	00_h	3	RECORD	r/w	—	Sende- PDO 4 Mappingparameter	—
	01-08_h	3	UINTEGER32	r/w	—	PDO Mapping Parameter für das nte Objekt	—
2014_h	00_h	S	ARRAY	r	—	Maske PDOs 37...40 Kanal 1	—
	01_h	S	UINTEGER32	r/w	—	Maske (Byte 0..3)	—
	02_h	S	UINTEGER32	r/w	—	Maske (Byte 4..7)	—
2015_h	00_h	S	ARRAY	r	—	Maske PDOs 37...40 Kanal 2	—
	01_h	S	UINTEGER32	r/w	—	Maske (Byte 0..3)	—
	02_h	S	UINTEGER32	r/w	—	Maske (Byte 4..7)	—
2016_h	00_h	S	ARRAY	r	—	Maske PDOs 37...40 Kanal 3	—
	01_h	S	UINTEGER32	r/w	—	Maske (Byte 0..3)	—
	02_h	S	UINTEGER32	r/w	—	Maske (Byte 4..7)	—
2017_h	00_h	S	ARRAY	r	—	Maske PDOs 37...40 Kanal 4	—
	01_h	S	UINTEGER32	r/w	—	Maske (Byte 0..3)	—
	02_h	S	UINTEGER32	r/w	—	Maske (Byte 4..7)	—
2020_h	00_h	S	RECORD	r	—	Lageregler	—
	01_h	S	UINTEGER8	r/w	—	Achstyp	POSCNFG
	02_h	S	INTEGER32	r/w	—	In-Positionsfenster	PEINPOS
	03_h	S	INTEGER32	r/w	—	Schleppfehlerfenster	PEMAX
	04_h	S	INTEGER32	r/w	—	Positionsregister 1	SWE1
	05_h	S	INTEGER32	r/w	—	Positionsregister 2	SWE2
	06_h	S	INTEGER32	r/w	—	Positionsregister 3	SWE3
	07_h	S	INTEGER32	r/w	—	Positionsregister 4	SWE4
	08_h	S	UINTEGER32	r/w	—	Auflösung Nenner	PGEARO
	09_h	S	UINTEGER32	r/w	—	Auflösung Zähler	PGEAR
	$0A_h$	S	UINTEGER8	r/w	—	Zählrichtung	DIR

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2022_h	00_h	S	RECORD	r	—	Positionierdaten für den Betriebsmode: Lage	—
	01_h	S	INTEGER32	r/w	☒	Position (Fahrsatz 0)	O_P
	02_h	S	UINTEGER16	r/w	☒	Geschwindigkeit gewichtet s. ASCII VMUL (Fahrsatz 0, s.a. SDO 2022_h Subindex$0D_h$)	O_V
	03_h	S	UINTEGER16	r/w	☒	Fahrauftragsart (Fahrsatz 0)	O_C
	04_h	S	INTEGER32	r/w	☒	Position für ext. Trajektorie	—
	05_h	S	UINTEGER16	r/w	☒	Fahrauftragsnummer	—
	06_h	S	UINTEGER16	r/w	☒	Anfahrtzeit (Fahrsatz 0)	O_ACC1
	07_h	S	UINTEGER16	r/w	☒	Bremszeit (Fahrsatz 0)	O_DEC1
	08_h	S	UINTEGER16	r/w	☒	Ruckbegrenzung für Anfahrtzeit (Fahrsatz 0)	O_ACC2
	09_h	S	UINTEGER16	r/w	☒	Ruckbegrenzung für Bremszeit (Fahrsatz 0)	O_DEC2
	$0A_h$	S	UINTEGER16	r/w	☒	Nummer des Folgefahrauftrages	O_FN
	$0B_h$	S	UINTEGER16	r/w	☒	Startverzögerung für Folgefahrauftrag	O_FT
	$0C_h$	S	2 x UINTEGER16	r/w	—	Kopieren eines Fahrauftrages	OCOPY
	$0D_h$	S	UINTEGER16	r/w	—	Wichtungsfaktor Geschwindigkeit (SDO 2022_h Sub-index 02_h)	VMUL
	$0E_h$	S	UINTEGER32	r/w	☒	Geschwindigkeit (Fahrsatz 0)	O_V
2024_h	00_h	S	RECORD	r	—	Einrichtbetrieb für den Mode „Lage“	—
	01_h	S	UINTEGER8	r/w	—	Referenzfahrtart	NREF
	02_h	S	UINTEGER8	r/w	—	Referenzfahrtrichtung	DREF
	03_h	S	INTEGER32	r/w	—	Geschwindigkeit Referenzfahrt	VREF
	04_h	S	UINTEGER16	r/w	—	Beschleunigungsrampe [Tippen & Referenzieren]	ACCR
	05_h	S	UINTEGER16	r/w	—	Bremsrampe [Tippen & Referenzieren]	DECR
	06_h	S	INTEGER32	r/w	—	Referenzoffset	ROFFS
	07_h	S	INTEGER32	r/w	—	Geschwindigkeit Tippbetrieb	VJOG
2026_h	00_h	S	RECORD	r	—	Sonderobjekte für den Mode Position	—
	01_h	S	UINTEGER8	r/w	—	Aktivierung der internen Auswerteeinheit für das Speichern des Positionsistwertes	—
2050_h	00_h	S	ARRAY	r	—	Hilfsvariablen der digitalen Eingänge	—
	01_h	S	INTEGER32	r/w	—	Triggervariable Eingang 1	IN1TRIG
	02_h	S	INTEGER32	r/w	—	Triggervariable Eingang 2	IN2TRIG
	03_h	S	INTEGER32	r/w	—	Triggervariable Eingang 3	IN3TRIG
	04_h	S	INTEGER32	r/w	—	Triggervariable Eingang 4	IN4TRIG
2051_h	00_h	S	ARRAY	r	—	Konfiguration für Schwellenregister	—
	01_h	S	UINTEGER32	r/w	—	Überwachung (deaktivieren / aktivieren)	WPOSE
	02_h	S	UINTEGER32	r/w	—	Meldeart (umlaufend / einmalig)	WPOSX
	03_h	S	UINTEGER32	r/w	—	Polarität für Positionsmeldung	WPOSP
2052_h	00_h	S	ARRAY	r	—	Positionschwellenregister ABSOLUT (Aktualisie-rungszeit < 1ms)	POSRSTAT
	01_h	S	INTEGER32	r	—	Positionregister	P1
	02_h	S	INTEGER32	r	—	Positionregister	P2
	03_h	S	INTEGER32	r	—	Positionregister	P3
	04_h	S	INTEGER32	r	—	Positionregister	P4
	05_h	S	INTEGER32	r	—	Positionregister	P5
	06_h	S	INTEGER32	r	—	Positionregister	P6
	07_h	S	INTEGER32	r	—	Positionregister	P7
	08_h	S	INTEGER32	r	—	Positionregister	P8
	09_h	S	INTEGER32	r	—	Positionregister	P9
	$0A_h$	S	INTEGER32	r	—	Positionregister	P10
	$0B_h$	S	INTEGER32	r	—	Positionregister	P11
	$0C_h$	S	INTEGER32	r	—	Positionregister	P12
	$0D_h$	S	INTEGER32	r	—	Positionregister	P13
	$0E_h$	S	INTEGER32	r	—	Positionregister	P14
	$0F_h$	S	INTEGER32	r	—	Positionregister	P15
	10_h	S	INTEGER32	r	—	Positionregister	P16

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2053_h	00_h	S	INTEGER8	r	—	Positionschwellenregister RELATIV (Aktualisierungs-zeit < 1ms) ASCII: s.a.	POSRSTAT
	01_h	S	INTEGER32	r	—	Positionregister	P1
	02_h	S	INTEGER32	r	—	Positionregister	P2
	03_h	S	INTEGER32	r	—	Positionregister	P3
	04_h	S	INTEGER32	r	—	Positionregister	P4
	05_h	S	INTEGER32	r	—	Positionregister	P5
	06_h	S	INTEGER32	r	—	Positionregister	P6
	07_h	S	INTEGER32	r	—	Positionregister	P7
	08_h	S	INTEGER32	r	—	Positionregister	P8
	09_h	S	INTEGER32	r	—	Positionregister	P9
	$0A_h$	S	INTEGER32	r	—	Positionregister	P10
	$0B_h$	S	INTEGER32	r	—	Positionregister	P11
	$0C_h$	S	INTEGER32	r	—	Positionregister	P12
	$0D_h$	S	INTEGER32	r	—	Positionregister	P13
	$0E_h$	S	INTEGER32	r	—	Positionregister	P14
	$0F_h$	S	INTEGER32	r	—	Positionregister	P15
	10_h	S	INTEGER32	r	—	Positionregister	P16
2060_h	00_h	S	INTEGER32	r/w	—	Drehzahl oder Stromsollwert	—
2070_h	00_h	S	RECORD	r	—	herstellerspez. Istwerte	—
	01_h	S	INTEGER24	r	⊠	Istlage (Drehwinkel, Auflösung: 20 Bit pro Umdrehung)	PRD
	02_h	S	INTEGER24	r	⊠	Drehzahl in Inkr. (1[Inkr] = 1875/262144 [Umd/min])	—
	03_h	S	INTEGER32	r	⊠	Inkr. Ist.-Position (Auflösung abhängig von PRBASE)	—
	04_h	S	UINTEGER16	r	⊠	gespeicherte Position (HW - LATCH positiv, 16 Bit pro Umdrehung)	LATCH16
	05_h	S	INTEGER32	r	⊠	gespeicherte Position (HW - LATCH positiv, 32 Bit pro Umdrehung)	LATCH32
	06_h	S	INTEGER32	r	⊠	Position in Abhängigkeit der Getriebefaktoren (PGEARI, PGEARO)	PFB
	07_h	S	INTEGER32	r	—	Geschwindigkeit in Abhängigkeit der Getriebefakto-ren (PGEARI, PGEARO)	PV
	08_h	S	INTEGER32	r	⊠	Schleppfehler in Abhängigkeit der Getriebefaktoren (PGEARI, PGEARO)	PE
	09_h	S	UINTEGER32	r	—	eff. Strom	I
	$0A_h$	S	INTEGER32	r	⊠	Drehzahl 1/min	V
	$0B_h$	S	INTEGER32	r	—	Kühlkörpertemperatur	TEMPH
	$0C_h$	S	INTEGER32	r	—	Innentemperatur	TEMPE
	$0D_h$	S	INTEGER32	r	—	Zwischenkreisspannung	VBUS
	$0E_h$	S	INTEGER32	r	—	Ballastleistung	PBAL
	$0F_h$	S	INTEGER32	r	—	I2t - Belastung	I2T
	10_h	S	INTEGER32	r	—	Betriebsdauer	TRUN
	11_h	S	INTEGER32	r	⊠	erw. Status für TPDO 33	—
2080_h	00_h	S	ARRAY	r	—	Statusinformationen	—
	01_h	S	UINTEGER8	r	⊠	Input/Output/Latch	—
	02_h	S	UINTEGER8	r	⊠	Sammelinfos (Warning/Error/Reserve)	—
	03_h	S	UINTEGER8	r	⊠	DRVSTAT LSB	DRVSTAT
	04_h	S	UINTEGER8	r	⊠	“	DRVSTAT
	05_h	S	UINTEGER8	r	⊠	“	DRVSTAT
	06_h	S	UINTEGER8	r	⊠	DRVSTAT MSB	DRVSTAT
	07_h	S	UINTEGER8	r	⊠	TRJSTAT LSB	TRJSTAT
	08_h	S	UINTEGER8	r	⊠	“	TRJSTAT
	09_h	S	UINTEGER8	r	⊠	“	TRJSTAT
	$0A_h$	S	UINTEGER8	r	⊠	TRJSTAT MSB	TRJSTAT
2081_h	00_h	S	INTEGER8	r	—	Inkr Ist. - Position (byteweise/24 Bit)	—
	01_h	S	INTEGER8	r	⊠	Inkr. Pos LSB	—
	02_h	S	INTEGER8	r	⊠	IstIncpos	—
	03_h	S	INTEGER8	r	⊠	IstIncpos	—
	04_h	S	INTEGER8	r	⊠	Inkrementelle Position MSB	—
	05_h	S	INTEGER24	r	⊠	Inkrementelle Position 24 Bit	—
	06_h	S	INTEGER32	r	⊠	Inkrementelle Position 32 Bit	—

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
2082$_h$	00$_h$	S	INTEGER8	r	—	Latch 32/24 Bit positiv Latch	—
	01$_h$	S	INTEGER32	r	☒	Latch 32Bit	LATCH32
	02$_h$	S	INTEGER24	r	☒	Latch 24Bit	—
2083$_h$	00$_h$	S	INTEGER8	r	—	Latch 32/24 Bit negativ Latch	—
	01$_h$	S	INTEGER32	r	☒	Latch 32Bit	LATCH32N
	02$_h$	S	INTEGER24	r	☒	Latch 24Bit	—
2084$_h$	00$_h$	S	INTEGER8	r	—	Latch 16 Bit positiv Latch	—
	01$_h$	S	INTEGER16	r	☒	Latch 16 Bit	LATCH16
2085$_h$	00$_h$	S	INTEGER8	r	—	Latch 16 Bit negativ Latch	—
	01$_h$	S	INTEGER16	r	☒	Latch 16 Bit	LATCH16N
2086$_h$	00$_h$	S	UINTEGER16	r	☒	Positionstriggerwort	POSRSTAT
2600$_h$	00$_h$	S	INTEGER8	r/w	—	1st Receive PDO select	—
2601$_h$	00$_h$	S	INTEGER8	r/w	—	2nd Receive PDO select	—
2602$_h$	00$_h$	S	INTEGER8	r/w	—	3rd Receive PDO select	—
2603$_h$	00$_h$	S	INTEGER8	r/w	—	4th Receive PDO select	—
2721$_h$	00$_h$	S	INTEGER8	r/w	—	Konfiguration Receive PDO 33	—
2A00$_h$	00$_h$	S	INTEGER8	r/w	—	1st Transmit PDO select	—
2A01$_h$	00$_h$	S	INTEGER8	r/w	—	2nd Transmit PDO select	—
2A02$_h$	00$_h$	S	INTEGER8	r/w	—	3rd Transmit PDO select	—
2A03$_h$	00$_h$	S	INTEGER8	r/w	—	4th Transmit PDO select	—
3100$_h$	00$_h$	S	Visible String	r/w	—	ASCII-Zeichen Richtung	—
3500$_h$	00$_h$	S	RECORD	r	—	Beginn des Objektkanals	—
	01$_h$	S	UNSIGNED16	r	—	Gesamtanzahl der Objekte im Objektkanal	MAXSDO
	02$_h$	S	UNSIGNED16	r	—	unterer Grenzwert	—
	03$_h$	S	UNSIGNED16	r	—	oberer Grenzwert	—
	04$_h$	S	UNSIGNED16	r	—	Defaultwert	—
	05$_h$	S	UNSIGNED8	r	—	Datenformat des Objektes	—
	06$_h$	S	UNSIGNED32	r	—	Kontrolldaten	—
	07$_h$	S	—	—	—	reserviert	—
	08$_h$	S	—	—	—	reserviert	—
3501$_h$	00$_h$	S	RECORD	r	—	Beschleunigungsrampe Drehzahlregelung	—
	01$_h$	S	UNSIGNED16	r	—	Wert	ACC
	02$_h$	S	UNSIGNED16	r	—	unterer Grenzwert	—
	03$_h$	S	UNSIGNED16	r	—	oberer Grenzwert	—
	04$_h$	S	UNSIGNED16	r	—	Defaultwert	—
	05$_h$	S	UNSIGNED8	r	—	Datenformat des Objektes	—
	06$_h$	S	UNSIGNED32	r	—	Kontrolldaten	—
	07$_h$	S	—	—	—	reserviert	—
	08$_h$	S	—	—	—	reserviert	—
6040$_h$	00$_h$	4	INTEGER16	w	☒	DS402 controlword	—
6041$_h$	00$_h$	4	INTEGER16	r	☒	DS402 Status	—
605A$_h$		4	INTEGER16	r/w	—	Quickstop option code	—
6060$_h$	00$_h$	4	INTEGER16	w	☒	Mode of Operation	—
6061$_h$	00$_h$	4	INTEGER16	r	☒	Display Mode of Operation	—
6063$_h$	00$_h$	4	INTEGER32	r	☒	Inkr. Istposition	—
6064$_h$	00$_h$	4	INTEGER32	r	☒	Istposition (unter Berücksichtigung der Getriebefaktoren)	—
606C$_h$	00$_h$	4	INTEGER32	r	☒	Istdrehzahl (Mode: pv)	—
607A$_h$	00$_h$	4	INTEGER32	r/w	☒	Zielposition (Mode: pp)	—
607B$_h$	00$_h$	4	ARRAY	r	—	position_range_limit	—
	01$_h$	4	INTEGER32	r/w	—	min_position_range_limit	—
	02$_h$	4	INTEGER32	r/w	—	max_position_range_limit	—
607C$_h$	00$_h$	4	INTEGER32	r/w	☒	Referenzoffset	—
6081$_h$	00$_h$	4	UINTEGER32	r/w	☒	Geschwindigkeit (Mode: pp)	—
6082$_h$	00$_h$	4	UNSIGNED32	r/w	☒	Endgeschwindigkeit (Mode: pp)	—
6083$_h$	00$_h$	4	UNSIGNED32	r/w	☒	Beschleunigung (Mode: pp pv)	—
6084$_h$	00$_h$	4	UNSIGNED32	r/w	☒	Bremsbeschleunigung (Mode: pp pv)	—
6086$_h$	00$_h$	4	INTEGER16	r/w	☒	Art der Fahrsatzrampen (motion_profil_typ)	—
6089$_h$	00$_h$	4	INTEGER8	r/w	☒	Notationsindex Position (wird nicht unterstützt)	—
608A$_h$	00$_h$	4	UNSIGNED8	r/w	☒	Dimensionsindex Position (wird nicht unterstützt)	—
608B$_h$	00$_h$	4	INTEGER8	r/w	☒	Notationsindex Drehzahl (wird nicht unterstützt)	—
608C$_h$	00$_h$	4	UNSIGNED8	r/w	☒	Dimensionsindex Drehzahl (wird nicht unterstützt)	—

Index	Sub-index	DEF	Datenformat	Zu-griff	PDO mappbar	Kurzbeschreibung	ASCII Objekt
$608D_h$	00_h	4	INTEGER8	r/w	⊠	Notationsindex Beschleunigung (wird nicht unterstützt)	—
$608E_h$	00_h	4	UNSIGNED8	r/w	⊠	Dimensionsindex Beschleunigung (wird nicht unterstützt)	—
6093_h	00_h	4	UNSIGNED32	r/w	—	Positionsfaktor	—
	01_h	4	UNSIGNED32	r/w	⊠	Numerator	PGEARO
	02_h	4	UNSIGNED32	r/w	⊠	Vorschubkonstante (feed_constant)	PGEARI
6094_h	00_h	4	UNSIGNED32	r/w	—	Geschwindigkeitsfaktor Encoder	—
	01_h	4	UNSIGNED32	r/w	⊠	Numerator	—
	02_h	4	UNSIGNED32	r/w	⊠	Divisor	—
6097_h	00_h	4	UNSIGNED32	r/w	—	Beschleunigungsfaktor	—
	01_h	4	UNSIGNED32	r/w	⊠	Divisor	—
	02_h	4	UNSIGNED32	r/w	⊠	Beschleunigungsfaktor	—
6098_h	00_h	4	INTEGER8	r/w	⊠	Referenzfahrtart	—
6099_h	00_h	4	ARRAY	r	—	Referenzfahrtgeschwindigkeit	—
	01_h	4	INTEGER32	r/w	⊠	Referenzfahrtgeschwindigkeit (bei Suche nach Referenzschalter)	—
$609A_h$	00_h	4	UNSIGNED32	r/w	⊠	Referenzfahrtbeschleunigung	—
$60FF_h$	00_h	4	INTEGER32	r/w	⊠	Drehzahlsollwert	—

VI.4 Neukonfiguration des SERVOSTAR

Im SERVOSTAR gibt es eine Anzahl von Parametern, die erst in der Initialisierungsphase beim Einschalten des Geräts in das laufende Firmware-Programm integriert werden und dann ihre Wirkung entfalten. Daher erfordert eine Änderung dieser Parameter ein Speichern aller Parameter und einen anschließenden Neustart des Geräts. Dies kann über den CAN-Bus durch folgende SDOs ausgelöst werden:

1. Schreiben des SDO $35EB_h$ Subindex 01_h, Wert 0, Speichern - Kommando
2. Schreiben des SDO 3632_h Subindex 01_h, Wert 0, Neustart des Geräts

Diese Seite wurde bewusst leer gelassen.

VI.5 Stichwortverzeichnis

Vertrieb und Service

Wir wollen Ihnen einen optimalen und schnellen Service bieten. Nehmen Sie daher bitte Kontakt zu der für Sie zuständigen Vertriebsniederlassung auf. Sollten Sie diese nicht kennen, kontaktieren Sie bitte den europäischen oder nordamerikanischen Kundenservice.

Europa

Besuchen Sie die europäische Danaher Motion Website auf **www.DanaherMotion.net**.
Dort finden Sie die aktuelle Inbetriebnahmesoftware, Applikationshinweise und die neuesten Produkthandbücher.

Danaher Motion Kundenservice - Europa

Internet **www.DanaherMotion.net**
E-Mail **virtapp@danaher-motion.net**
Tel.: +49(0)203 - 99 79 - 0
Fax: +49(0)203 - 99 79 - 155

Nordamerika

Besuchen Sie die nordamerikanische Danaher Motion Website auf **www.DanaherMotion.com**.
Dort finden Sie die aktuelle Inbetriebnahmesoftware, Applikationshinweise und die neuesten Produkthandbücher.

Danaher Motion Customer Support North America

Internet **www.DanaherMotion.com**
E-Mail **customer.support@danahermotion.com**
Tel.: (815) 226 - 2222
Fax: (815) 226 - 3148

